Real-Time Analysis on YouTube for Trend Recognition

San Jose State University

Hamsalakshmi Ramachandran, 017423666,
Himani Shah, 017411407,
Soumya Challuru Sreenivas, 017518618,
Prithvi Elancherran, 017518709

***Abstract*—With the incremental popularity of YouTube as an universal content platform being the second largest search engine, comes the streams of humongous volumes of data, emphasizing the importance of trend analysis for business and creators in real time.The objective of the project "Real-Time Analysis on YouTube for Trend Recognition" is to develop an effective solution for large volumes of big data with the aim of finding new trends in YouTube's content and researching these trends. For this, the system makes use of the YouTube API to automatically extract video descriptions, user comments, and the number of views, likes and dislikes, and user interactions with the content. Adding several technologies, the platform offers news about trending content, illustration's dominant keywords, and several engagements altogether, however, near real time they are filed. This project addresses the difficulties associated with the processing of rapidly changing non – structured data, using state of the art big data technology and methodology. It shows the possibility of performing real-time analysis of extensive databases generated within social networks. It elucidates how raw data can be processed into information for optimal decision making.**

***Index Terms*—YouTube trends, Apache Kafka, Spark Streaming, Explainable AI, Differential Privacy, Locality Sensitive Hashing (LSH).**

I. Introduction

THE exponential growth of online video platforms like YouTube generates a continuous data stream, making real-time trend analysis essential for marketers, content creators, and researchers. Traditional batch processing systems fail to provide actionable insights in a timely manner. This paper proposes a real-time framework for analyzing YouTube trends by integrating streaming algorithms, privacy-preserving techniques, and explainable AI. Our approach ensures scalability, privacy, and interpretability, making it suitable for dynamic data environments.

II. Motivation

YouTube generates massive amounts of user data daily, making it a valuable resource for identifying trends and understanding viewer behavior. However, traditional batch processing systems are insufficient for analyzing this continuously evolving data in real time. There is a pressing need for a system that can efficiently process high-throughput data streams while addressing critical challenges like scalability, privacy, and interpretability. Sensitive user data such as views, likes, and comments must be analyzed without compromising privacy, and AI-driven analysis must be transparent to foster trust. This project tackles these challenges by leveraging stream processing frameworks, such as Kafka and Spark, to analyze data , integrating Differential Privacy to protect sensitive information, and using Locality Sensitive Hashing (LSH) for clustering similar videos. By combining these techniques with Explainable AI tools, the project delivers a scalable, privacy-preserving, and transparent solution for real-time YouTube trend analysis.

III. Literature Review

A. Understanding YouTube Trends

In YouTube Trend Analysis [1], Arushi Pathik et al. explore the factors that make a video trending on YouTube. By employing machine learning algorithms like Linear Regression, Decision Trees, and Gaussian Naive Bayes, the study provides actionable insights for content creators aiming to boost video popularity. Through the analysis of trending and non-trending videos, key metrics such as views, likes, comments, and upload time are identified as pivotal to virality. The authors offer a comprehensive guide for optimizing these parameters, enabling creators to maximize audience engagement and reach.

B. Evaluating Apache Kafka for Real-Time Data Streaming

Shubham Vyas et al. present Performance Evaluation of Apache Kafka: A Modern Platform for Real-Time Data Streaming [2], investigating Kafka's capabilities as a distributed platform for real-time data processing. The study examines throughput, latency, polling intervals, and partition configurations, offering insights into Kafka's efficiency in managing large data volumes. Results highlight Kafka's robustness in high-volume, low-latency scenarios, showcasing its scalability, fault tolerance, and suitability over traditional batch processing methods.

C. Innovations in Dynamic Bloom Filters

In The Dynamic Bloom Filters [3], Deke Guo et al. propose an advanced Bloom filter that accommodates dynamic datasets. Unlike traditional static Bloom filters, the dynamic version supports operations like insertion, deletion, and union while maintaining a low false-positive rate. The study features a detailed mathematical analysis and performance evaluation, demonstrating its memory efficiency and applicability in dynamic environments. This innovation is particularly beneficial for distributed systems requiring space-efficient, adaptable data structures.

D. Explainable AI in Security Operations

Håkon Svee Eriksson et al., in Towards XAI in the SOC: A User-Centric Study of Explainable Alerts with SHAP and LIME [4], delve into the role of explainable artificial intelligence (XAI) in Security Operation Centers (SOC). Using SHAP and LIME methods, the study helps SOC analysts interpret machine learning-generated alerts by explaining the features contributing to these alerts. Interviews with analysts revealed the tools' potential to enhance trust and usability, underlining the importance of XAI in improving decision-making in cybersecurity.

IV. Development Methodology

To track the advancement of our work, we utilized Trello to organize tasks and monitor progress effectively. The Trello board for this project is available at Link. Pair programming was adopted as our primary development methodology, which facilitated efficient teamwork and improved time management during the project's development and testing phases. As a result, we concluded that the project output greatly benefited from the integration of this collaborative approach. Discussions and brainstorming sessions were conducted via Zoom meetings, while Google Sheets and Docs were employed for real-time collaboration and documentation. Each team member actively participated by alternating roles as the driver and observer, ensuring equal contribution and a well-rounded understanding of the project.

V. System Architecture Overview

This system is designed to perform real-time trend analysis on YouTube data by leveraging big data technologies for ingestion, processing, privacy preservation, and explainability. Below are the key components and their roles:

A. Data Ingestion

The YouTube Data API fetches video data, which is ingested into the system through Apache Kafka, a robust message broker. Kafka ensures low-latency data transmission and fault tolerance by streaming the data into distributed topics, enabling scalable processing.

B. Stream Processing

Kafka serves as the backbone of the project's real-time data pipeline. It is used to stream video data from the YouTube API to the processing and storage components. The Kafka Producer collects and pushes data (e.g., video details like title, views, likes, and hashtags) to the Kafka topic `youtube_topic`. The Kafka Consumer continuously reads this data and processes it using PySpark.

1) Kafka Configuration Details:

- **Topic Name:** `youtube_topic`
- **Partitions:** 6 (ensuring parallelism and scalability for real-time data streaming).
- **Retention Policy:** Messages are retained for 1 week, allowing time for consumers to process the data.
- **Max Message Size:** Configured to 2 MB (2097164 bytes), ensuring larger payloads (e.g., detailed video data) can be transmitted seamlessly.
- **Cleanup Policy:** Set to delete, meaning messages are automatically removed after the retention period expires.

Kafka was chosen for its ability to handle high-throughput, low-latency streaming data efficiently. It ensures reliability through message partitioning, replication, and fault tolerance, making it ideal for real-time applications like analyzing YouTube video trends. Kafka also allows seamless scaling as the volume of streaming data grows, maintaining the robustness of the system.

C. Privacy Integration

Differential Privacy techniques are applied to sensitive data (e.g., views, likes, and comments). Noise is added to ensure individual data cannot be traced while still allowing accurate aggregate insights. This step complies with ethical standards for data usage in large-scale systems.

D. Locality Sensitive Hashing (LSH)

Videos are clustered by similarity in hashtags or content, enabling categorization and analysis of related trends. This technique efficiently groups videos in high-dimensional spaces, helping to identify communities or patterns within the data.

E. Explainability

The system uses SHAP (SHapley Additive exPlanations) to provide transparency into how features like hashtags, upload time, and engagement metrics influence predictions or trends. This empowers users to understand the driving factors behind the results.

F. Insight Generation and Delivery

The processed and clustered data is used to generate metrics and visualizations for trend analysis. Insights such as trending topics, audience preferences, and hashtag performance are delivered to users through dashboards or reports, offering valuable data for decision-making.

VI. Design Steps

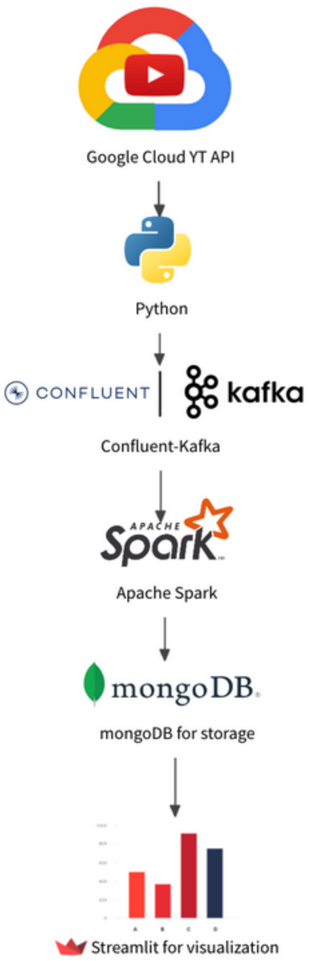


Fig. 1. System Architecture for Real-Time Trend Analysis

VII. Algorithms and Techniques

A. Bloom Filter

In this project, we have used Bloom filter to ensure the uniqueness of hashtags extracted from YouTube video descriptions. After extracting hashtags using regular expressions, each hashtag is checked against the Bloom filter to determine if it has already been processed. Unique hashtags are added to the filter and included in the final data, while duplicates are ignored. This step is integrated into the Kafka producer to optimize the real-time processing of video data.

The Bloom filter is chosen for its memory efficiency and constant-time performance, making it ideal for handling large volumes of data in real time. Unlike traditional data structures like lists or sets, it uses less memory and scales well with high data throughput.

B. DGIM

Streaming algorithms, specifically DGIM (Datar-Gionis-Indyk-Motwani), are employed to calculate approximate counts for metrics like views, likes, and comments within a sliding window of the last hour. This algorithm dynamically updates the counts by aggregating incoming data into compact ”buckets” while discarding older data beyond the sliding window. DGIM’s efficiency makes it suitable for real-time scenarios where continuously storing all individual data points is impractical.

In this project, DGIM is used in the consumer script to track the total views, likes, and comments over time without consuming excessive memory. These sliding window metrics are critical for understanding real-time trends and ensuring the system can report on the most recent activity accurately.

C. LSH

LSH (Locality-Sensitive Hashing) is used to cluster hashtags based on their semantic similarity, aiding in trend discovery and topic grouping. Hashtags are converted into numerical vectors using TF-IDF (Term Frequency-Inverse Document Frequency), which quantifies their importance in the dataset. LSH maps these vectors into buckets, grouping similar ones together by hash similarity, without requiring exhaustive pairwise comparisons.

This clustering process is implemented in the consumer script, where hashtags are extracted from video descriptions and processed. The output clusters represent related or trending topics, helping the system identify connections between content themes and allowing for advanced trend analysis.

D. Differential Privacy

Differential privacy is incorporated to anonymize sensitive metrics like views, likes, and comments of videos. These metrics, while crucial for trend analysis, are considered sensitive as they can reveal details about individual content creators and their engagement. Laplace noise is added to these values to mask individual contributions, ensuring that the aggregated results are privacy-preserving while remaining useful for analysis.

By applying differential privacy, the project anonymizes metrics before they are used for insights such as top-trending videos, categories, and hashtags. This protects the data of content creators, ensuring their engagement statistics are not exposed or misused.

E. Implementing Explainable AI

In our project, Explainable AI (XAI) is implemented using SHAP (SHapley Additive exPlanations) to interpret the clustering results of hashtags. After grouping hashtags into clusters using LSH and TF-IDF, SHAP is applied to explain the importance of features (like specific terms in hashtags) that contributed to these groupings. By generating visualizations like SHAP summary and force plots, the project provides a clear understanding of why certain hashtags are clustered together, making the process transparent.

VIII. Results

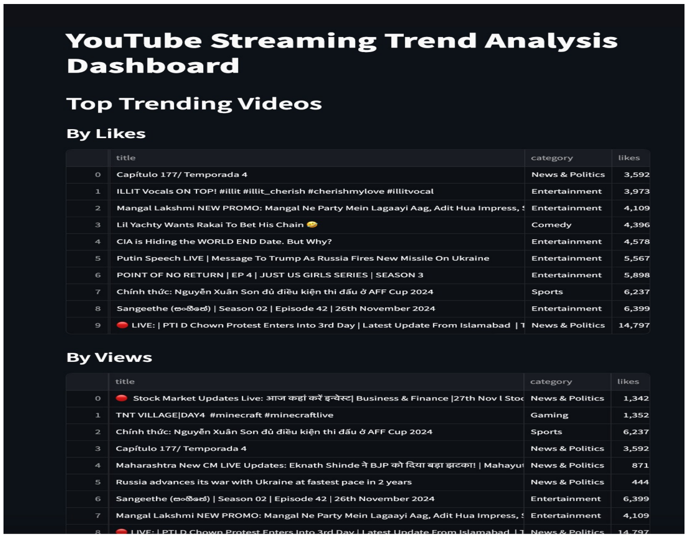
IX. Streamlit Dashboard and Visualizations

The project used Streamlit to build an interactive dashboard, providing real-time visualizations of trending videos, hashtags,

TABLE I
STEPS, TASKS, AND TOOLS USED

Step	Task Performed	Tools Used
1	Choosing a topic & its scope	Zoom meetings
2	Proposal	Google Docs, Zoom meetings for collaboration
3	Dataset	YouTube API for real-time video metadata collection
6	Data Storage	MongoDB (NoSQL) and JSON files for intermediate storage
7	Data Cleaning & ETL	Python (Pandas), PySpark for filtering, transformations
8	Streaming Tools	Kafka for real-time ingestion and Spark Streaming for processing
9	Algorithms	Bloom filter, DGIM, LSH, Differential Privacy, and SHAP
10	Visualization	Streamlit for generating visualization dashboard
12	Version Control	GitHub repository for code management and collaboration
13	Slides	Gamma web app
14	Report	Google Docs, Overleaf (LaTeX template) for structured report writing

and user engagement metrics. The dashboard connects seamlessly to MongoDB to display processed data effectively.



YouTube Streaming Trend Analysis Dashboard

Top Trending Videos

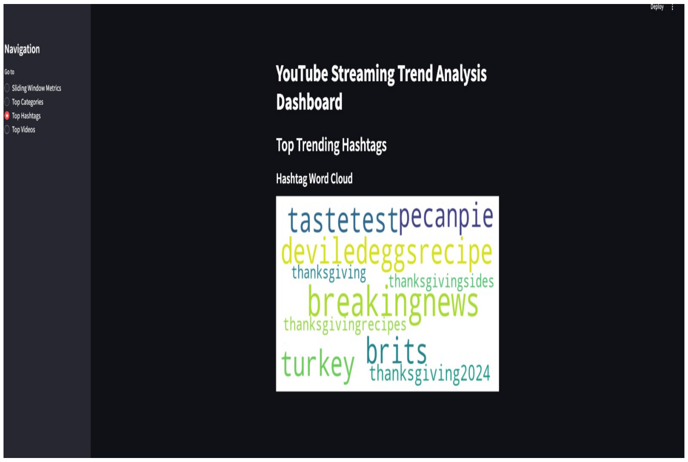
By Likes

	title	category	likes
0	Capítulo 177/ Temporada 4	News & Politics	3,592
1	ILLIT Vocals ON TOP! #illit #illit_cherish #cherishmylove #illitvocal	Entertainment	3,973
2	Mangal Lakshmi NEW PROMO: Mangal Ne Party Mein Lagaayi Aag, Adit Hua Impress, S	Entertainment	4,109
3	Lil Yachty Wants Rakai To Bet His Chain 🤣	Comedy	4,396
4	CIA is Hiding the WORLD END Date. But Why?	Entertainment	4,578
5	Putin Speech LIVE | Message To Trump As Russia Fires New Missile On Ukraine	Entertainment	5,567
6	POINT OF NO RETURN | EP 4 | JUST US GIRLS SERIES | SEASON 3	Entertainment	5,898
7	Chính thức: Nguyễn Xuân Son đủ điều kiện thi đấu ở AFF Cup 2024	Sports	6,237
8	Sangeethe (සංගීතේ) | Season 02 | Episode 42 | 26th November 2024	Entertainment	6,399
9	🔴 LIVE: | PTI D Chown Protest Enters Into 3rd Day | Latest Update From Islamabad | T	News & Politics	14,797

By Views

	title	category	likes
0	🔴 Stock Market Updates Live: आज कहां करें इन्वेस्ट| Business & Finance |27th Nov l Stoc	News & Politics	1,342
1	TNT VILLAGE|DAY4 #minecraft #minecraftlive	Gaming	1,352
2	Chính thức: Nguyễn Xuân Son đủ điều kiện thi đấu ở AFF Cup 2024	Sports	6,237
3	Capítulo 177/ Temporada 4	News & Politics	3,592
4	Maharashtra New CM LIVE Updates: Eknath Shinde ने BJP को दिया बड़ा झटका! | Mahayut	News & Politics	871
5	Russia advances its war with Ukraine at fastest pace in 2 years	News & Politics	444
6	Sangeethe (සංගීතේ) | Season 02 | Episode 42 | 26th November 2024	Entertainment	6,399
7	Mangal Lakshmi NEW PROMO: Mangal Ne Party Mein Lagaayi Aag, Adit Hua Impress, S	Entertainment	4,109
8	🔴 LIVE: | PTI D Chown Protest Enters Into 3rd Day | Latest Update From Islamabad | T	News & Politics	14,797

Fig. 2. Streaming Trend Analysis Dashboard

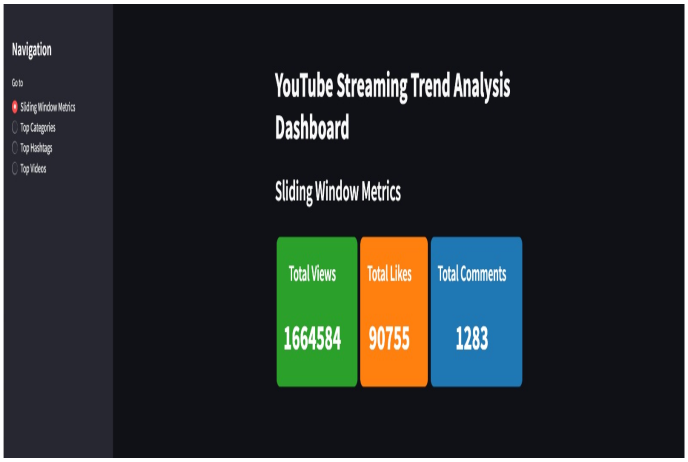


YouTube Streaming Trend Analysis Dashboard

Top Trending Hashtags

Hashtag Word Cloud

Fig. 3. Top Trending Hashtags

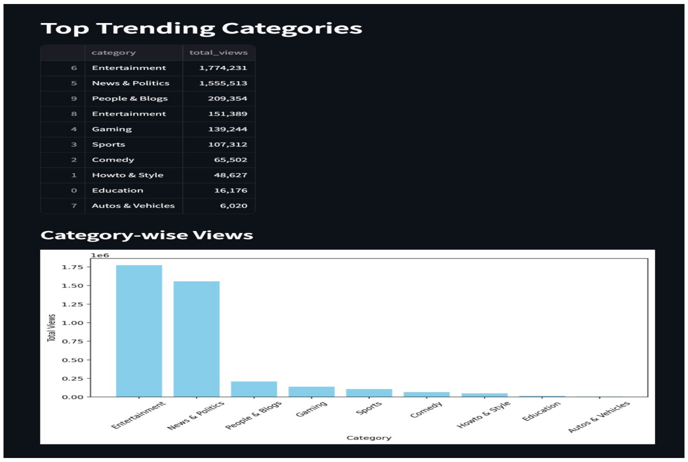


YouTube Streaming Trend Analysis Dashboard

Sliding Window Metrics

Total Views	Total Likes	Total Comments
1664584	90755	1283

Fig. 4. Sliding Window Metrics



Top Trending Categories

	category	total_views
6	Entertainment	1,774,231
5	News & Politics	1,555,513
9	People & Blogs	209,354
8	Entertainment	151,389
4	Gaming	139,244
3	Sports	107,312
2	Comedy	65,502
1	Howto & Style	48,627
0	Education	16,176
7	Autos & Vehicles	6,020

Category-wise Views

Fig. 5. Top Trending Categories

X. Technical Difficulties

A. Handling API Quotas and Errors

Working with the YouTube Data API required managing daily request limits and implementing robust error-handling mechanisms. This taught us how to optimize API usage while ensuring data continuity. We faced YouTube API usage limitation as it has a daily limit (10,000 units/day).

B. Difficulty in Integrating Producer and Consumer

Integrating the producer and consumer on Confluent with the SASL_SSL protocol for secure connections posed significant challenges. Ensuring secure communication while maintaining functionality required detailed configuration and troubleshooting.

C. Issues with Bucketed Random Projection LSH

Bucketed Random Projection LSH struggled with generating meaningful clusters, especially when the dataset included a large number of videos. Fine-tuning parameters to improve the clustering performance required additional effort.

D. Data Transfer Challenges to MongoDB

Transferring processed data from the Kafka consumer to MongoDB faced data insert issues. Debugging these issues required careful analysis of both the data schema and MongoDB configurations.

E. Handling Sensitive Credentials

Handling sensitive credentials for Kafka and MongoDB in the codebase raised security concerns. Implementing proper credential management techniques, such as environment variables and secret managers, was necessary to address these issues.

XI. Key Learnings

A. Real-Time Data Processing

Experience in setting up Kafka producers and consumers for real-time ingestion and analysis, understanding how Confluent Kafka and Spark Streaming efficiently process high-throughput data streams.

B. Tool Integration and Workflow Automation

Gained expertise in integrating tools such as YouTube Data API, Confluent Kafka, PySpark, and MongoDB to create a seamless workflow. Successfully automated the entire pipeline, from data collection to final analysis, leveraging intervals and Spark transformations for efficient and streamlined processing.

C. How to Choose and Prioritize Different Algorithms in Workflow

Acquired in-depth knowledge of cutting-edge algorithms, including DGIM (Datar-Gionis-Indyk-Motwani) for efficient sliding window computations of metrics like views and likes, Bloom Filters for fast and memory-efficient membership testing, Bucketed Random Projection LSH (Locality Sensitive Hashing) for clustering similar hashtags in high-dimensional spaces, and Differential Privacy for protecting sensitive data while preserving analytical utility. Gained valuable experience in assessing and selecting algorithms tailored to the project's specific goals and challenges.

D. Privacy Preservation

Implemented Differential Privacy by introducing Laplace noise to sensitive metrics such as views, likes, and comments, effectively safeguarding user data. This approach ensured a balance between protecting individual privacy and maintaining the accuracy of analytical insights, promoting ethical and responsible data handling.

E. The Role of Explainable AI

Gained a comprehensive understanding of the importance of Explainable AI in real-world applications. Learned how tools like SHAP ensure transparency by making complex analyses, such as clustering and ranking of videos, interpretable for human decision-makers. Recognized its role in fostering trust when working with sensitive user data and its ability to detect biases in features or algorithms. Additionally, learned how insights from Explainable AI can be used to improve models by refining feature selection and ensuring compliance with ethical AI principles for accountable and transparent data processing.

F. Collaborative Development and Paired Programming

Collaborative development played a crucial role in the project's success. Agile methodologies, supported by Trello boards for task management, ensured accountability and team visibility. Pair programming enhanced code quality and knowledge sharing, while Zoom meetings facilitated effective brainstorming, progress reviews, and problem-solving. GitHub Copilot boosted productivity through intelligent code suggestions, and GitHub's version control enabled seamless collaboration. Together, these tools streamlined workflows, improved communication, and ensured a well-coordinated and successful project. Regular sprints ensured progress was monitored and goals were met on time.

G. System Design Principles

The project provided valuable experience in designing modular and scalable architectures tailored for real-time data processing. It involved learning how to seamlessly integrate multiple components, including APIs, Kafka, Spark, MongoDB, and visualization tools, into a cohesive and efficient workflow. This approach ensured the system was robust, flexible, and capable of handling large-scale data streams effectively.

H. Visualization and Reporting

The project demonstrated the importance of clear data visualization for effective communication, leveraging tools like Streamlit and MongoDB Charts to present key insights such as trending videos and hashtags. Expertise was gained in creating interactive dashboards and optimizing data pipelines for efficient and user-friendly real-time visualizations.

XII. Version Control

We uploaded our work in GitHub with public viewing. Anyone can check out our project.
Link: https://github.com/Himani03/Youtube-trend-analysis.git

XIII. Conclusion

With the exponential growth of online video content and the demand for real-time insights, analyzing YouTube data becomes crucial for understanding trends and audience behavior. This project leverages Apache Kafka, an open-source publisher/subscribe messaging system, to handle high-throughput, real-time video data streams efficiently. Data is collected using the YouTube API, processed using PySpark, and stored in MongoDB for visualization in Streamlit. The dynamic nature of video streaming rates requires scalable and efficient processing. To address this, we implemented advanced algorithms such as DGIM for sliding window metrics and LSH for clustering, ensuring robust performance under varying data loads. Differential privacy safeguards sensitive metrics like views, likes, and comments, while Explainable AI (SHAP) provides interpretability for clustering outcomes. This project demonstrates a comprehensive and scalable framework for real-time video trend analysis, offering actionable insights while preserving user privacy and maintaining system transparency.

References

1) A. Pathik et al., ”YouTube Trend Analysis,” Proceedings of the 2022 International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST), 2022.
2) Shubham Vyas et al., ”Performance Evaluation of Apache Kafka: A Modern Platform for Real-Time Data Streaming,” Big Data Analytics, vol. 10, pp. 45–56, 2023.
3) Deke Guo et al., ”The Dynamic Bloom Filters,” Distributed Systems Journal, vol. 18, pp. 789–798, 2023.
4) Håkon Svee Eriksson et al., ”Towards XAI in the SOC: A User-Centric Study of Explainable Alerts with SHAP and LIME,” Cybersecurity AI Journal, vol. 7, pp. 345–360, 2023.
5) Databricks, ”Processing Data in Apache Kafka with Structured Streaming in Apache Spark 2.2,” Databricks Blog, April 2017.
6) KeesTalksTech, ”Streaming a Kafka topic in a Delta table on S3 using Spark Structured Streaming,” KeesTalksTech Blog, November 2019.

TABLE II
Rubrics for Project Evaluation

Criteria	**Comments**	**Points**
Code Walkthrough	Code is in GitHub and walk through will be done during presentation	5
Presentation Skills	Time Management Evaluation by professor	5
Discussion / Q&A	During presentation	5
Demo	During presentation	5
Report	Comprehensive report submitted including architecture, methodologies, streaming algorithms, differential privacy, and innovation	7
Version Control	https://github.com/Himani03/Youtube-trend-analysis.git	3
Lessons Learned	Included in report	5
Prospects of Winning Competition / Publication	Included in report	3
Innovation	Implemented techniques such as SHAP for explainable AI, differential privacy for data security, and LSH for clustering hashtags.	5
Teamwork	Members and roles included in report	5
Technical Difficulty	Included in report	5
Practiced Pair Programming	Included in report (Development Methodology)	2
Practiced Agile / Scrum (1-week sprints)	With the use of Trello	2
Used Grammarly / Other Tools for Language	Grammarly used for error-checking and improving report quality	1
Slides	During presentation and added in GitHub	3
Used LaTeX	Overleaf using LaTeX for report formatting (IEEE template). .tex file included in GitHub	2
Used Creative Presentation Techniques	Used Gamma tool for ppt	2
Literature Survey	Included in report	7
Use of Streaming Algorithms	Used Bloom Filter and DGIM	5
Use of Stream Processing Frameworks	Used Kafka and PySpark.	5
Use of Locality-Sensitive Hashing	Included in the code.	5
Use of Privacy Techniques	Used Differential Privacy	5
Any Other Tools and Techniques	Used Explainable AI - SHAP	5
Use of New Tools Not Included in HW	Used Confluent. Utilized MongoDB for data storage	3