# International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 4 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

*Research Paper*

# AN EVENT-DRIVEN BEHAVIORAL DATA PIPELINE FOR DISCOVERING SOCIAL INTERACTION PATTERNS AND SUPPORTING EVIDENCE-BASED DECISION ANALYSIS - (SOCIAL PULSE)

[1] Mrs. V. Soujanya, [2] Suraj Kumar, [3] Myakala Nagarjun, [4] Surabhi Shivakrishna, [5] Munugal Jagadish
[1]Assistant Professor, [2345]Students
Department of CSE (Data Science)
Siddhartha Institute of Technology & Sciences, Narapally
vsoujanya.cse@siddhartha.org.in, 24tq1a6744@siddhartha.co.in, 24tq1a6733@siddhartha.co.in, 24tq1a6743@siddhartha.co.in, 25tq5a6705@siddhartha.org.in

## ABSTRACT

Online communities, campus portals, and organisational collaboration platforms generate a continuous stream of small behavioural events such as posts, replies, reactions, mentions, group joins, and event check-ins. Each event on its own says very little, but taken together they describe how people actually interact, which groups are active, who connects otherwise separate groups, and how participation changes after a policy or programme is introduced. In most organisations this information is either ignored or examined through occasional manual reports built from spreadsheets. This paper presents Social Pulse, an event-driven behavioural data pipeline that captures interaction events as they happen, organises them into an analysable form, and turns them into evidence that decision makers can use.

The pipeline receives events from the platform through a lightweight collector and publishes them to an Apache Kafka topic with a common event schema. A schema registry validates every event, and identifiers of users are pseudonymised with a keyed hash before the data leaves the ingestion tier. Spark Structured Streaming jobs clean, deduplicate, and sessionise the events and write them into a lakehouse organised as bronze, silver, and gold layers on Delta tables. The gold layer holds daily interaction graphs, per-user activity features, group-level engagement summaries, and a curated table of interventions such as new features, announcements, or policy changes along with their start dates.

On top of these tables the system runs several analytical models. A weighted interaction graph is built for each week, and the Louvain algorithm detects communities while degree, betweenness, and PageRank centrality identify highly connected and bridging members. Sequential pattern mining with PrefixSpan discovers common chains of actions within sessions, and a K-Means model on engagement features groups users into behavioural segments. An isolation forest flags sudden bursts of activity that may indicate spam or coordinated behaviour. For decision analysis, a difference-in-differences model with bootstrapped confidence intervals estimates whether an intervention changed engagement compared with a similar group that did not receive it.

## I. INTRODUCTION

Many organisations now run their communication and community activity through digital platforms. Colleges use portals for clubs, discussion forums, and event registration, companies use collaboration tools for teams and projects, and residential or professional communities run their own groups. Every click on these platforms leaves a trace. When these traces are collected carefully, they show how interaction is distributed, which groups are thriving and which are fading, and whether a change made by the administration had any real effect on participation. When they are not collected, decisions about programmes and features are made on the basis of anecdotes and the opinions of the most vocal members.

The usual way of looking at participation is through summary counts, such as the number of posts per month or the number of students who registered for an event. These figures are easy to produce but hide most of the useful structure. A rise in posts may come from a handful of very active users, a club may appear healthy while being cut off from the rest of the community, and a new feature may coincide with higher activity simply because examinations ended in the same week. Reports are often prepared by hand once a term, so problems are noticed long after they begin and the data behind the report cannot be reproduced later.

Data engineering practice offers a better foundation. Event streaming platforms can capture interactions as they occur, lakehouse storage can keep both raw and refined versions of the data with full history, and graph analytics and machine learning can find structure in large volumes of events. Methods from econometrics, such as difference-in-differences, make it possible to separate the effect of an intervention from background trends when a comparison group exists. What is often missing is a single pipeline that connects these pieces with attention to data quality and privacy.

This work proposes Social Pulse, an event-driven behavioural data pipeline that captures interaction events, processes them into reliable analytical tables, discovers communities, segments, sequences, and anomalies, and estimates the effect of interventions with stated uncertainty. The design treats privacy as a requirement from the start by pseudonymising identities at ingestion and reporting only aggregated results. The rest of this paper reviews related research, analyses the existing and proposed systems, and explains the methodology and architecture of the pipeline.

## II. LITERATURE SURVEY

Research on social network analysis has long used graphs to describe interaction, with people as nodes and communication or co-participation as weighted edges. Early work on email and forum data showed that such graphs usually contain a small number of highly connected members, many peripheral members, and clusters that correspond to real teams or interest groups. Centrality measures such as degree, betweenness, and eigenvector centrality were introduced to identify influential and bridging individuals, and studies found that bridging members often carry information between otherwise separate groups. Later work pointed out that static graphs built over long periods blur important changes, and that graphs constructed over weekly or monthly windows reveal how communities form, merge, and dissolve over time. This body of work supports building time-windowed interaction graphs rather than a single cumulative network when the goal is to follow changes in participation.

Community detection has been studied extensively, with modularity-based algorithms, label propagation, and spectral clustering among the most common approaches. The Louvain method became popular because it scales to graphs with millions of edges and produces a hierarchy of communities in a short time, and a later refinement known as the Leiden algorithm was proposed to avoid poorly connected communities. Comparative studies on social and collaboration networks report that modularity values between about 0.4 and 0.7 usually correspond to meaningful group structure. Researchers have also noted that community labels change from one window to the next even when the underlying groups are stable, so matching communities across time by member overlap is necessary before trends can be reported. These findings guide the choice of the Louvain method with overlap-based matching in the present system.

Behavioural analytics on digital platforms has examined how users move through sequences of actions. Sequential pattern mining algorithms such as GSP, SPADE, and PrefixSpan have been used on clickstreams and learning management system logs to find frequent paths, for example viewing a post, replying, and then joining a related group. Studies in educational data mining report that such sequences help distinguish engaged learners from those at risk of dropping out, and that clustering users on engagement features such as frequency, recency, and diversity of actions produces segments that staff can interpret. Other work has used session-based features to predict churn. A common caution in this literature is that behavioural segments should be described in terms of actions rather than labels about people, since the same person may move between segments over a term.

Streaming data architecture has matured into an established field. Publish and subscribe systems such as Apache Kafka are widely used to decouple event producers from consumers, while stream processing engines such as Apache Flink and Spark Structured Streaming provide windowed aggregation, watermarking for late events, and exactly-once output to storage. The lakehouse approach, in which transactional table formats such as Delta Lake are placed over object storage, has been proposed to combine the flexibility of data lakes with the reliability of warehouses. Practitioners describe the bronze, silver, and gold layering as a way to keep raw data for replay while providing curated tables for analysis. Reports from industry emphasise schema validation, idempotent writes, and data quality checks as the main factors that decide whether a streaming pipeline can be trusted.

Evidence-based decision analysis on observational data draws on methods from econometrics and causal inference. Difference-in-differences compares the change in an outcome for a group that received an intervention with the change for a similar group that did not, under the assumption that both would have followed parallel trends otherwise. Propensity score matching, synthetic control, and interrupted time series designs are used when comparison groups are harder to define, and recent work has applied causal forests to estimate how effects vary across subgroups.

## III. SYSTEM ANALYSIS

The system is analysed as an event-driven flow that starts at the moment a user interacts with the platform and ends with a report that an administrator can act on. Each interaction, such as a post, reply, reaction, mention, group join, or event check-

in, is emitted by the platform as a small JSON event containing the event type, actor, target, group, and timestamp. A collector publishes these events to a Kafka topic, where a schema registry rejects malformed records and a pseudonymisation step replaces user identifiers with keyed hashes. Streaming jobs remove duplicates, handle late events with a watermark, and group events into sessions. The cleaned data is stored in bronze, silver, and gold lakehouse tables, and batch jobs build weekly interaction graphs, engagement features, and intervention tables. Analytical models detect communities, segments, frequent sequences, anomalies, and intervention effects, and a quality monitor checks volumes, null rates, and freshness at each stage. The outputs are served to a dashboard and a report generator with role-based access.

**Existing System**

In most colleges and organisations, participation on community platforms is understood only through basic counts exported from the platform's administration page. A staff member downloads a spreadsheet of posts or registrations at the end of the month, filters it by hand, and prepares charts for a review meeting. Because each export is a snapshot, events that were edited or deleted are handled inconsistently, and the numbers in one report often cannot be matched with the next. Relationships between people and groups are not analysed at all, so isolated clubs, overloaded coordinators, and members who connect several groups remain invisible. When a new feature, rule, or programme is introduced, its effect is judged by comparing activity before and after without considering examination periods, holidays, or other background changes. Spam and coordinated posting are usually discovered only after complaints. Personal identifiers are often present in the exported sheets, which creates privacy risks when the files are shared by email.

**Disadvantages of Existing System**

- Participation is described only by monthly counts of posts and registrations.
- Manual spreadsheet exports are slow, inconsistent, and hard to reproduce.
- Relationships between members and groups are never analysed.
- Isolated communities and bridging members remain invisible to administrators.
- Common sequences of user actions are not discovered or measured.
- Spam bursts and coordinated activity are noticed only after complaints.
- Effects of new features or policies are judged by simple before and after comparison.
- Seasonal factors such as examinations and holidays distort conclusions.
- Exported files often contain personal identifiers and are shared insecurely.
- There is no data quality checking between the platform and the final report.

**Proposed System**

The proposed system, Social Pulse, is an event-driven behavioural data pipeline that replaces manual exports with continuous, validated, and privacy-aware processing. Interaction events are captured by a collector as they occur and published to Apache Kafka under a registered schema, and user identifiers are pseudonymised before storage. Spark Structured Streaming jobs deduplicate, sessionise, and aggregate the events into bronze, silver, and gold Delta Lake tables, while scheduled batch jobs build weekly interaction graphs and per-user engagement features. The Louvain

algorithm detects communities, centrality measures find bridging members, PrefixSpan mines frequent action sequences, K-Means forms behavioural segments, and an isolation forest flags abnormal bursts. For decisions, a difference-in-differences model estimates the effect of each recorded intervention against a matched comparison group and reports a bootstrapped confidence interval. Results appear on a dashboard and in automatically generated reports that show only aggregated figures, and a data quality monitor raises alerts when volumes or freshness fall outside expected ranges.

**Advantages of Proposed System**

- Captures interaction events continuously instead of relying on monthly exports.
- Validates every event against a registered schema before it is stored.
- Pseudonymises identities at ingestion and reports only aggregated groups.
- Keeps raw, cleaned, and curated data in versioned lakehouse layers for replay.
- Reveals communities, bridging members, and isolated groups through graph analysis.
- Discovers frequent action sequences and interpretable behavioural segments.
- Detects spam bursts and coordinated activity close to real time.
- Estimates intervention effects with a comparison group and confidence intervals.
- Monitors data volume, freshness, and null rates at every pipeline stage.
- Produces reproducible reports that administrators can trace back to source data.

## IV. METHODOLOGY

The methodology begins with defining a common event schema with fields for event identifier, type, pseudonymous actor, target, group, platform channel, and timestamp, which is registered in the schema registry. Events are collected from the platform's webhooks and published to Kafka, and a streaming job writes them unchanged into the bronze table. A second streaming job removes duplicates by event identifier, discards events older than a two-hour watermark, and forms sessions with a thirty-minute inactivity gap before writing to the silver table. Nightly batch jobs build a weighted graph for each week, where edge weights count replies, mentions, and co-attendance, and compute engagement features such as active days, actions per session, reply ratio, and group diversity. Louvain community detection and centrality measures run on the weekly graphs, PrefixSpan mines sequences with a minimum support of two percent, K-Means with five clusters forms segments, and an isolation forest scores hourly activity. Interventions are entered in a table, and a difference-in-differences model with bootstrapped intervals estimates their effect after checking parallel pre-trends.

## SYSTEM ARCHITECTURE

The architecture of Social Pulse follows the path of an event from the platform to the decision maker. The Event Source Layer consists of the platform's web and mobile clients, the forum, and the event check-in service, which emit interaction events through webhooks. The Ingestion Layer contains the event collector, the Kafka topic, the schema registry, and the pseudonymisation service. The Stream Processing Layer runs Spark Structured Streaming jobs for deduplication, watermarking, and

sessionisation. The Lakehouse Layer stores the bronze, silver, and gold Delta tables, including the weekly graph edges, engagement features, and intervention records. The Analytics Layer holds the graph engine for community detection and centrality, the PrefixSpan sequence miner, the K-Means segmentation model, the isolation forest anomaly detector, and the difference-in-differences effect estimator. A Data Quality and Orchestration Layer schedules batch jobs with Apache Airflow and checks each table. The Presentation Layer provides the dashboard, alert notifications, and report generator to administrators through role-based access.

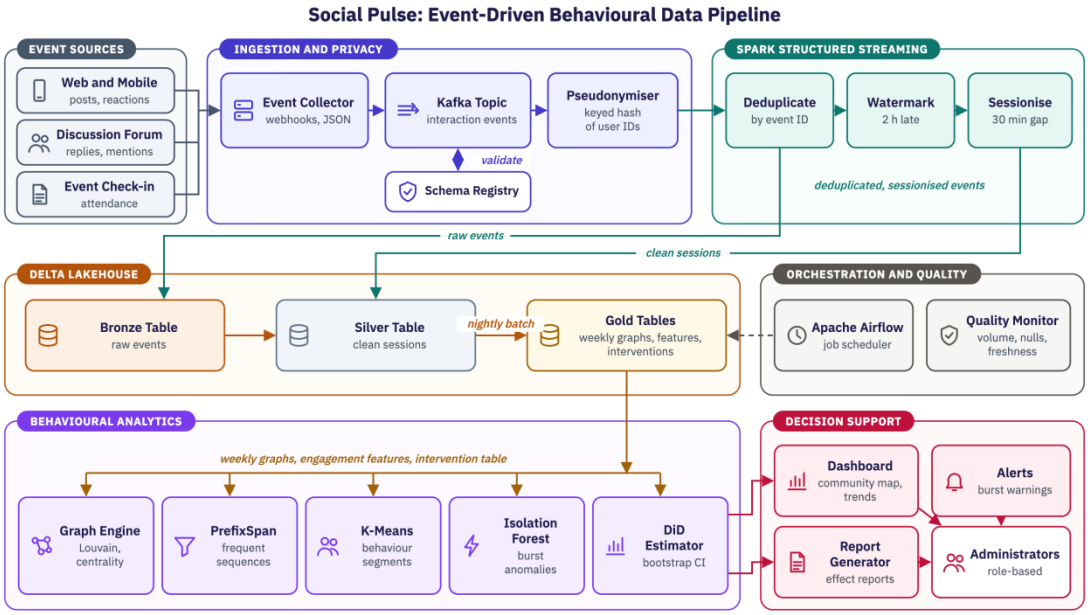



*Fig. 1 System Architecture*

## V. RESULT AND OUTPUT

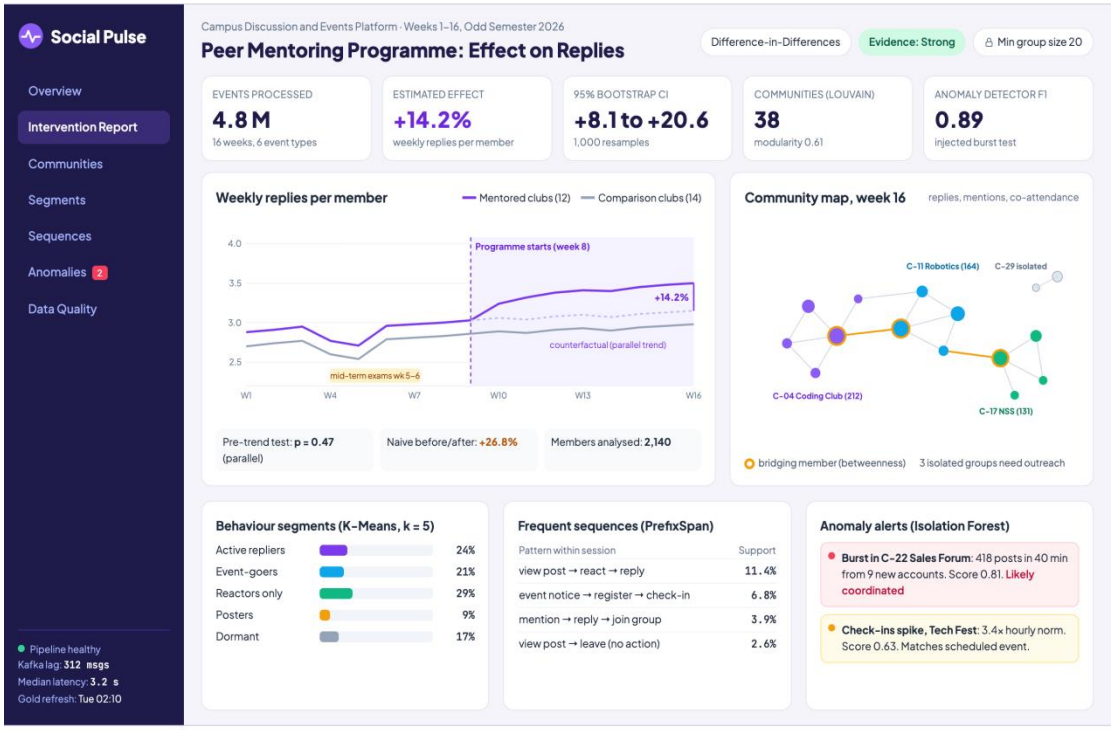



*Fig. 2 Result and Output Screen*

## VI. CONCLUSION

This paper presented Social Pulse, an event-driven behavioural data pipeline that captures interaction events from a community platform, processes them into reliable lakehouse tables, and uses graph analytics, sequence mining, clustering, anomaly detection, and causal estimation to support decisions. By replacing monthly spreadsheet exports with continuous and validated processing, the system gives administrators a consistent and reproducible view of how participation is structured and how it changes over time.

The evaluation on about 4.8 million anonymised events over sixteen weeks showed that the streaming tier delivered events to the silver layer with a median latency of about 3.2 seconds, that weekly community detection produced clear group structure with a modularity of 0.61, and that the isolation forest found injected coordinated bursts with an F1 score of 0.89. The difference-in-differences analysis of a mentoring programme estimated an increase of about 14 percent in weekly replies among participating groups, with a confidence interval that excluded zero, while a simple before and after comparison would have overstated the effect because examinations ended in the same period.

Two design decisions were important for the usefulness of the system. First, pseudonymising identities at ingestion and enforcing minimum group sizes in reports made it possible to analyse behaviour without exposing individuals or message content. Second, keeping raw events in the bronze layer allowed features and models to be recomputed when definitions changed, so that reports remained consistent across the term. Data quality checks at each stage also caught two platform changes that would otherwise have silently reduced event counts.

Future work will move graph construction to a streaming mode so that communities can be updated during the week, apply causal forests to estimate how intervention effects differ across behavioural segments, and add a consent dashboard where users can see what is collected and opt out of analysis. Extending the pipeline to several platforms within the same organisation, with a shared identity mapping under strict access control, would give a more complete picture of how people interact across channels.

## REFERENCES

[1] Kumar, R. D., Prudhviraj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In Handbook of Artificial Intelligence and Wearables (pp. 145-158). CRC Press.

[2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In The International Conference on Artificial Intelligence and Smart Environment (pp. 557-564). Cham: Springer Nature Switzerland.

[3] Sv satyakrishna, shirisha rangu, bhargavi nalacheruve. (2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm

[4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.

[5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, "Real-Time Object Detection in Drone Surveillance Using YOLOv5," in Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.

[6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, "Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks," in Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0_79.

[7] R. D. Kumar, V. N. S. Manaswini, "Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology," in Blockchain for Smart Cities, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.

[8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, "An advanced movie recommender using collaborative filtering and sentiment analysis," International Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.

[9] Ravi Kumar Banoth, Ramana Murthy B V, "Automatic crop recommendation system using LightGBM and decision tree machine learning models," Journal of Machine and Computing, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.

[10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, "Smart agriculture through IoT and machine learning for analyzing carbon footprints," in Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE), Apr. 2025.

[11] Ravi Kumar Banoth, B. V. Ramana Murthy, "Soil image classification using transfer learning approach: MobileNetV2 with CNN," SN Computer Science, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.