INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 5, May 2026 | ISSN: 2456-4184 | IJNRD.ORG

REAL-TIME DATA STREAMING WITH ANOMALY DETECTION AND ALTERING

1st Nithish Ragu
Department of Advanced Computing & Analytics Vels Institute of Science, Technology & Advanced Studies
Chennai,India

2nd Dr.G.Thailambal
Department of Advanced Computing & Analytics Vels Institute of Science, Technology & Advanced Studies
Chennai,India

Abstract-Organizations generate huge amount of data from transactions, user interactions and system activities. In traditional methods the data are collected and stored first and then analyze it using batch processing techniques. Because of this, they cannot provide the real-time insights or alerts which cause delays in detecting critical issues such as fraud, sudden drops in sales or inventory shortages. To design and implement a real time streaming, monitoring and alerting using python and GCP services. A data generator is developed by python to produce live e-commerce data and send it to Google cloud pub/sub. A Python Subscriber process this data in real time and analyzes it instantly and it use rule-based anomaly detection techniques to detect unusual patterns. If an unusual or abnormal activity is detected the system immediately sends an alert via email. All analyzed data including anomalies are stored in BigQuery for monitoring and reporting. It helps organizations to monitor continuously and respond quickly for critical issues and take better decisions by using real-time data. The project illustrate how real-time streaming and alerting can help to improve efficiency, reliability and performance of Business in modern environments.

INTRODUCTION

Each second an e-commerce generator produce huge amount of data from customers orders and transactions. To understand the customer behavior and observe the daily operations firm needs to monitor the data that are generated. In traditional systems the responses for critical issues may be delayed. Because that process data after collected since it is slow and do not provide immediate insights. By analyzing data immediately as it produced, real time data streaming technologies assist in solving this problem. Cloud technologies.

provide efficient processing and storage of real-time e-commerce data. Because of their scalability and dependability systems helps to manage large transaction volume while preserving operation and data flow. Anomaly detection and alerting are integrated into streaming process to improve real time monitoring. Unusual patterns like increase or decrease in sales or questionable transactions are found with the help of anomaly detection. In order to facilitate ongoing monitoring and quick decision- making the project focus on improving a real-time e-commerce data streaming system with Python and Google Cloud services.

LITERATURE SURVEY

The rise in the number of e-commerce organizations and data because of a high rate of transactions, "real time anomaly detection" was proposed as a research concept. The inefficiency in handling this problem in "batch processing systems" led to the proposal of "real time systems." Many researchers have proposed the use of data processing techniques such as "distributed systems," "cloud computing," "message systems," etc., in order to keep a track of transactions. The aim of this system is to detect abnormal trends such as an increase in transactions, bulk transactions, etc. Many techniques have been proposed to detect these abnormalities, as proposed in various publications. The concept of anomaly detection is

an important tool in fraud detection, finance management, and system reliability in the field of e-commerce. In reference to the management of large quantities of data in the form of streams, various research studies have emphasized the importance of scalability in providing data warehouses and real-time analytics via the cloud platform. However, existing techniques are not able to address the issue of integration in data generation, visualization, and alerting. Recently proposed techniques are based on providing data ingestion, processing, storage, and visualization in a single platform to address this issue. With the integration of real-time streaming anomaly detection rules, and a dashboard in transaction management, the proposed project is similar to recent development.

PROPOSED WORK

By integrating a real-time data monitoring and alerting, the recommended solution seeks to address the drawbacks of the current system. This system continuously collect and analyze streaming data from the sources like sales transactions and user behavior in variation of batch processing systems. It use rule-based anomaly detection to automatically spot the unusual trends, sudden increases or decreases in sales that can cause any problems. The system immediately send an alerts via email when the abnormal pattern is found.The solution decreases the need for manual monitoring, reduce the possibility of late replies and make companies to active, aware decisions by integrating automatic alerting with real-time monitoring. In addition, the recommended solution provide practical insights through reports, assisting companies in spotting patterns, streamlining processes and boosting general productivity. Organizations can sustain seamless operations, enhance decision-making processes and react quickly to change in their data environment. The proposed system will enhance the capacity of organizations to continuously observe their operations without waiting for the processing of the data. Through the analysis of data organizations will easily able to detect any abnormal patterns of transaction. This helps the organization to act quickly and minimize the loss of money. Another significant advantage of this proposed system is it has the ability to handle large amount of data. This system has able to manage continuous flow of transactions without any challenges with the help of cloud based technology.Additionally the integration of automated anomaly detection and the features of alerting the need for manual monitoring is also diminishes. Instead of manual monitoring the system itself detects the suspicious activities and send an alerts to the admin. This help the organization to overcome the issues in time and never miss any issues. Every data that are processed by the system will be stored for further analysis. Thus the proposed system offers a reliable and efficient way for conducting business.

DESIGN SYSTEM

The use case diagram represent the flow of operations in real time streaming, observing and alerting system. The process starts with the data producer that produce transactional data. This data are fed in streaming system like pub/sub where the streaming data is processed and transferred and thedata is processed by the analytics engine. In this phase the data will also validated and transformed into a useful format.After the data is processed, it will be sent to detect the anomaly that is find the unusual patterns in the data. Once the data is processed that will be stored for future analysis. If any anomaly is detected the alerting system is activated to send alerts to the admin. This entire process is designed to help continuously monitoring the data in real time detecting anomaly and acting in a timely manner.

Fig 1. Use case diagram

Fig 2. Dataflow diagram

Fig 3. Architecture diagram

PROJECT PHASES / METHODOLOGIES

The Implementation of the real time streaming and anomaly detection model was performed in multiple stages to make sure the structured developed process like testing and deployment. Every stage focused on specific component of the model, including generating data, analyzing, development and testing.

PHASE 1: DATA COLLECTION

The first phase concentrated on generation of the data for the purpose of simulation in real time data streaming. Since we cannot obtain the real time e-commerce data for the privacy and the security reason we used a python based data generation tool to replicate the real time e-commerce transaction data. The data which we are generated is similar to the real time online transactional data and that also generated in time intervals to replicate the real time data.

PHASE 2: MODEL DEVELOPMENT

In model development phase the main processing logic of the model and the anomaly detection process are designed and implemented. In the project real time data streaming was implemented and developed a model with a rule based anomaly detection. We used a python subscriber to receive the transaction data from the google cloud pub/sub. Each transaction data is processed when it receives the generated data. This helps in analyzing the transaction data without waiting for the large amount of data that has been sent by Google cloud pub/sub.

PHASE 3: UI DEVELOPMENT

An interface is created in this phase and it is simple and easy for users to understand the monitoring of real time data streaming, detecting anomalies and sending alerts. It is mostly based on python programming and cloud based services and it expect the interface will be simple, easy to understand and functional for users. The main objective of the interface is to allow users to identify transactions quickly and any abnormal activities without needing broad knowledge of the process.

The key features of the interface are:

- Display python console: it shows the real time streaming data with the details of order ID, product name, price, quantity and total amount.
- Detecting anomalies: indicates the data as normal or anomaly.
- Alert notification: it send immediate alert through email when the anomaly are detected.
- Interface of data viewing: use Google bigquery for access the stored transaction records and anomaly.

about the generated data and they can get the output easily in the system. This user interface has made it possible for non-technical users to use the system properly.

PHASE 4: TESTING & DEPLOYMENT

The last phase of the project all the modules were developed are integrated, tested and deployed to ensure that the system are effectively in a real time environment. The goal of this stage of the modeldevelopment life cycle was to confirm that all the components of the system were working properly in individually and also in together. The entire flow of generating of data to detect the anomaly, send a alert notification and storage of data are verified to confirm functioning of the data.

IMPLEMENTATION

System integration:The process of system integration is all the modules were developed in various stages of system development cycle were integrated. Every module of the system plays an important role in functioning the model for implementation of the real time streaming and monitoring the model. The system integration ensured that flow of data between the various modules of the system that are remained unbroken. The various components of integrated system includes:

- Python data generator – it generates the e- commerce transactions.
- Google cloud pub/sub – it acts as a messaging service to send the data between the components.
- Python subscriber – it use to receive the data from the pub/sub.

- Anomaly detection module – to detect any abnormal pattern in the transactions.
- Alert system – to send instant alert for any abnormal transaction detected.
- Google bigquery – to store the processed transactions and any abnormal transactions that are detected.

The python data generator continuously generates transactional data like product names, categories, quantity, price, etc. Those data are published to the pub/sub service which act as the real time platform for the data source and processing module. The python subscriber then get the streaming data from the pub/sub and performs the required processing and validates the data before applying the anomaly detection logic.

Testing:To confirm that the system works correctly and reliably on various testing scenarios are carried out. The testing phase of the system development cycle is primarily focused on testing the functionality of the system. Different types of transactions data are created to replicate real world transactions.

The system tested under various scenarios:

- Normal transactional data to represent regular customer transactions.
- Used an high transactional data to test the anomaly detection.
- Constant flow of transactional data for long periods of time.
- Sudden rise in transactional value to replicate the abnormal activity.

These are the scenarios that are tested and used to verify the functionality of the system to handle the data correctly. The response of the system has been monitored in the testing phase.

After processing the data the model used rule based anomaly detection techniques to detect the anomalies of each transactions. If the transactions are found to be unusual pattern or abnormal activities then the system identity them as anomalies. When the abnormal transaction is detected immediately the system sends an alert to the admin.

The key goals of testing are:

- Confirm that the data publishing the accurate data to Google cloud pub/sub.
- Validate the receipt of the data in real time using python subscriber.
- Validate the accuracy of the anomaly detection based on the rules.
- Validate the instant notification of alerts for the anomalies.
- Validate the accuracy storage of processed data in Google BigQuery.

The test results are shown that the model is efficiently processed the generating data and detecting the abnormal transaction without any delays. Immediate alerts were sent for the detected anomalies, reflecting the effectiveness of the monitoring system.

Deployment:Once the test was successful the system was implemented in a cloud-based setup to ensure continuous operation in real time. This confirm that the system will be operate in a continuous flow, processing data streams without any problems.Additionally during the deployment of system a considerable focus was given to confirm the continuous performance and reliability of the system in real time. Particular focus has given to ensure that the system could operate effectively with large amount of data and minimal delay during the processing.

The key objectives of deployment are:

- Maintenance of stability within the system in continuous execution.
- Real time data processing with lower time of delay.
- Continuous streaming of transaction data.
- Reliable delivery of alerts.

Through the deployment process the system has successfully demonstrated its capability to monitor the data stream of the transactions and detect anomalies in the data stream instantly and provide alerts to the admin. This provides the capability of the proposed approach to support real time monitoring applications with the context of modern business world.

In conclusion the testing and deployment phase of proposed system has successfully validated the capability of the system to operate effectively and reliability within the context of the proposed solution. The proposed approach offers the ability to observe the transactional data within the organization continuously and respond to any risk within the system immediately.

SCREEN SHOTS

Fig 4. Data generator

Fig 5. Topic ID

Fig 5. Subscription ID

Fig 6. Orders Table

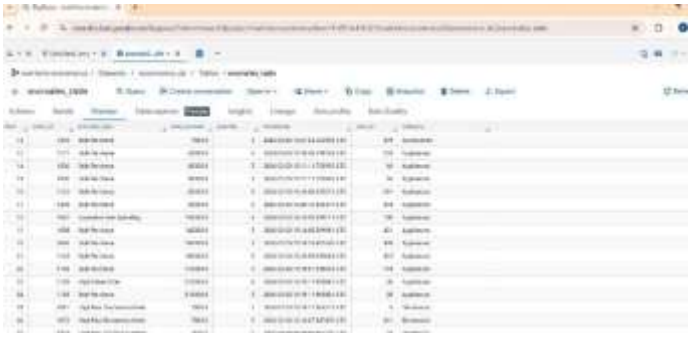


Fig 7. Anomaly Table

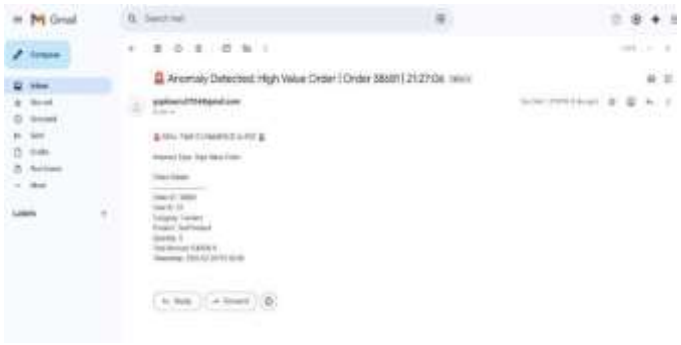
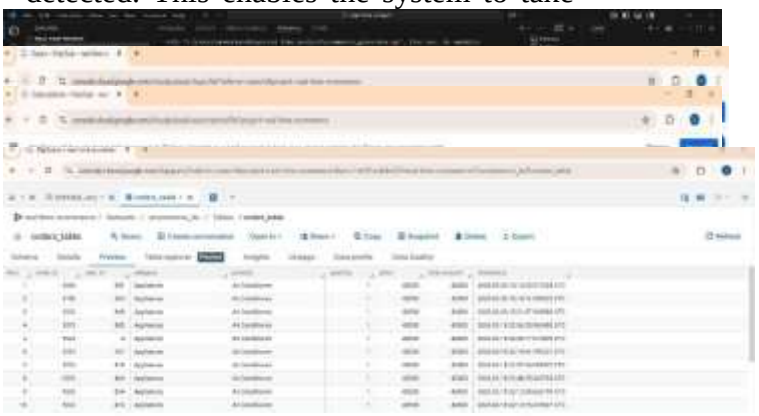


Fig 8. Alert Email

detected. This enables the system to take



immediate action in case of abnormal activity. By conducting various tests through different testing methods like unit testing, integration testing, user acceptance testing and performance evaluation of the system the system is verified to operate accurately. The tests have verified the system to operate effectively in handling the stream of data, accurately identifying anomalies in the data, sending immediate notifications in case of any abnormality and storing the data in the cloud environment.It can be concluded that the project is a huge success in achieving the goal of creating system for real time monitoring and anomaly detection. This system will help to improve decision-making by providing instant insight into the transactional data. This system will helps to minimize the risks of any issue by sending the instant alert if any anomaly detected. The system will enhance the efficiency of the system. Moreover, the system will allow for further development by integrating analytics tools, machine learning techniques and predictive analysis.

REFERENCE

JOURNAL REFERENCE

Conclusion

The Real-time data streaming with anomaly detection and alerting was successfully designed and implemented to overcome the limitations of batch processing in system. It has been observed that, in many business environments, the conventional batch processing systems are not very effective in identifying unusual transactions because the data processing happens only at specific intervals of time. The proposed project has overcome the limitations by processing the data continuously and in real time that helps to detect unusual patterns and abnormal transactions.The proposed project has been developed using the cloud based technologies to make the system efficient and reliable. Python as been used to generate the data, pub/sub has been used to stream the data between the components of the system. Finally the data are stored in google bigquery which helps to store the data and process the huge amount of data. This data processing has designed to detect high value of transaction and unusual activities that helps to identify risks during the data processing itself.Another significant component of the system is the alert notification system. This system provides immediate notification if any anomaly

[1] Chandola, V., Banerjee, A., & Kumar, V. "Anomaly detection: A survey." ACM Computing Surveys, vol. 41, no. 3, 2009, pp. 1–58. https://doi.org/10.1145/1541880.1541882

[2] Akidau, T., Bradshaw, R., Chambers, C., et al. "The Dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing." Proceedings of the VLDB Endowment, vol. 8, no. 12, 2015, pp. 1792–1803. https://doi.org/10.14778/2824032.2824076

[3] Zaharia, M., Das, T., Li, H., Shenker, S., & Stoica, I. "Discretized streams: Fault-tolerant streaming computation at scale." Proceedings of the ACM Symposium on Operating Systems Principles, 2013, pp. 423–438. https://doi.org/10.1145/2517349.2522737

[4] Laptev, N., Amizadeh, S., & Flint, I. "Generic and scalable framework for automated time-series anomaly detection." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1939–1947. https://doi.org/10.1145/2783258.2788611

[5] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 413–422, 2008. https://doi.org/10.1109/ICDM.2008.17

[6] Ahmed, M., Mahmood, A. N., & Hu, J. "A survey of network anomaly detection techniques." Journal of Network and Computer Applications, vol. 60, 2016, pp. 19–31. https://doi.org/10.1016/j.jnca.2015.11.016

[7] Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. "Unsupervised real-time anomaly detection for streaming data." Neurocomputing, vol. 262, 2017, pp. 134–147. https://doi.org/10.1016/j.neucom.2017.04.070

[8] Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. "Apache Flink: Stream and batch processing in a single engine." IEEE Data Engineering Bulletin, vol. 38, no. 4, 2015, pp. 28–38.

WEB REFERENCES

[1] Google Cloud."Pub/Sub Documentation." Available https://cloud.google.com/pubsub/docs

[2] GoogleCloud"BigQueryDocumentation."Availa ble:https://cloud.google.com/bigquery/docs

[3] Streamlit, “Streamlit Documentation.”

Available: https://docs.streamlit.io/

[4] Python Software Foundation. Available: https://www.python.org/
[5] SendGrid, "SendGrid Email API Documentation." Available: https://docs.sendgrid.com/
[6] Apache Spark, "Spark Streaming Guide."

Available: https://spark.apache.org/streaming/

[7] Scikit-learn, "Scikit-learn User Guide."

Available: https://scikit-learn.org/stable/

[8] Python, "Python Official Documentation."

Available: https://docs.python.org/3/