Real-Time Smart Bank Data Streaming Capture

Group - 03

Sindhu Nagesha (017419987)[b], Prayag Nikul Purani (017416737)[b], Syed Faraaz Ahmed (017428619)[b], Sai Vivek Chunduri (017435301).[b]

[a]*Department of Applied Data Science San Jose State University, DATA 228: Big Data Tech and Applications.*
[b]*Submitted to: Vishnu S Pendyala.*

28 April, 2024

Abstract

In the rapidly evolving landscape of digital banking, the demand for real-time data processing has become paramount in delivering personalized and responsive services to customers. This project presents the development and implementation of a real-time data analytics pipeline tailored specifically for the banking sector. The data is gathered from across multiple sources sensor, transactional and application. A robust technological stack, comprising Docker for containerization, Kafka for data streaming, and Apache Spark for real-time data processing, forming the foundation of the pipeline, ensuring efficient data ingestion and analysis. Implementation of IOT and timestream serves as a streamlined process flow for Sensor data. For transactional and application data Apache Kafka leverages its capabilities, the streamed data is processed swiftly, enabling prompt analysis and the derivation of actionable insights. Integration with Flask, and Elastic search facilitates instantaneous visuals providing notifications and alerts, empowering banking personnel to promptly address critical events such as fraud detection or transaction anomalies. Through the deployment of this intelligent bank data pipeline, organizations stand to gain invaluable insights into customer behavior, promptly identify fraudulent activities in real time, and deliver tailored banking experiences. This project underscores the transformative potential of real-time data analytics in revolutionizing the banking sector, ultimately enhancing customer satisfaction and loyalty levels.

1. Introduction

In today's digital era, data stands as the cornerstone of modern banking operations, driving efficiency and innovation. The convergence of rapid technological advancements and evolving customer expectations has propelled financial institutions into a realm where real-time access to actionable insights is indispensable. This project marks a pivotal journey to redefine how banks leverage data. We will be implementing data in two workflows: the first flow shows the sensor data which is connected to Amazon IOT core where the connected devices securely interact with the cloud applications and other devices. Timestream automates the data from the memory store to the magnetic store by connecting through the API and AWS Lambda. Here a logic is implemented based on which fraud transactions are captured. After this it is integrated with python where conversion of data, use of streaming algorithms is implemented and connection to Grafana is made that can be viewed as visuals in the application. The second workflow is implemented using application and transactional data where the data is sent and connection is made through python before streaming the data into Kafka where fraud detection logic is implemented. Apache sessions are created and tokenization and LSH algorithms are implemented before storing it in mongodb and then visualizing it in Grafana. And the other way being directly visualizing through the Flask to detect fraud. The ability to capture, process, and leverage data instantaneously empowers banks to detect patterns, anticipate trends, and make precise decisions swiftly. Moreover, real-time smart bank data streaming capture holds the potential to transform customer experiences, enabling hyper-personalized services and enhancing engagement and loyalty. By proactively identifying and addressing issues such as fraud or disruptions, banks can cultivate trust and long-term relationships with customers. This project signifies more than a technological upgrade; it signifies a strategic shift towards a data-centric banking approach to reshape the industry landscape. Embracing real-time smart bank data streaming capture opens new avenues for innovation, competitiveness, and sustainable growth in a dynamic marketplace.

2. Literature Survey

1. Apache Spark: A Big Data Processing Engine Eman Shaikh, Iman Ahmed Mohiuddin, Yasmeen Alufaisan and Irum Nahvi (2019).

In this paper, we learn how Big data refers to an excessively large amount of datasets that are used to computationally reveal patterns and trends. To analyze and find knowledge from this bulk of data, a processing framework is required. There are various types of commonly used big data frameworks such as Apache Hadoop, Apache Storm, Apache Spark, Apache Flink, etc. In this paper, we learn about Apache Spark's batch processing and stream processing abilities, use cases, ecosystem, architecture, multi-threading, and concurrency capabilities, and lastly the use of Spark in emerging technologies.

2. Information Security in Big Data: Privacy and Data Mining Xu, L., Jiang, C., Wang, J., Yuan, J. and Ren, Y. (2014).

Data mining can extract valuable knowledge and patterns from large datasets but also raises privacy concerns about sensitive personal information being disclosed. Privacy-Preserving Data Mining (PPDM) is a research area focused on modifying data to enable effective data mining while protecting sensitive information. Most PPDM work has looked at reducing privacy risks in the data mining operations phase. However, privacy threats can arise in other phases like data collection, publishing, and delivery of mining results. The paper takes a broader perspective, identifying four types of users involved in data mining applications: data provider, data collector, data miner, and decision maker. For each user type,the paper discusses their privacy concerns and methods to protect sensitive information throughout the knowledge discovery process. In Addition to reviewing privacy-preserving approaches per user role, the paper also covers game theoretical approaches that analyze interactions and valuations of sensitive information among different users. The goal is to provide insights into PPDM by differentiating the privacy responsibilities of different user roles in safeguarding sensitive information throughout the data mining pipeline.

3. Beyond Batch Processing: Towards Real-Time and Streaming Big Data Shahrivari, S. (2014).

This paper examines the limitations of traditional batch processing systems like Hadoop MapReduce in handling real-time queries, interactive jobs, and continuous data streams. It reviews emerging solutions for real-time processing, such as in-memory computing platforms and real-time query engines, as well as dedicated stream processing frameworks like Storm and S4. Through experimental results, the paper demonstrates the performance advantages of these new solutions over Hadoop for real-time and streaming workloads. The author concludes that while batch processing with Hadoop is mature, in-memory computing approaches like Spark are becoming essential to meet the growing real-time and streaming needs of big data applications.

4. Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges. F. Gürcan and M. Berigel.

This paper outlines the importance of real-time processing in today's technology landscape, especially in big data applications. It introduces a lifecycle for real-time big data processing, covering phases like data ingestion, storage, stream processing, analytical data store, and analysis/reporting. The document explores various tools such as Flume, Kafka, Nifi, Storm, Spark Streaming, S4, Flink, Samza, Hbase, Hive, Cassandra, Splunk, and Sap Hana, associating them with different lifecycle stages. Additionally, it addresses challenges like handling large and diverse data, ensuring consistency, scalability, real-time processing, data visualization, skill requirements, and privacy/security. The paper aims to provide insights into the lifecycle, tools, and challenges of real-time big data processing.

5. KAFKA:The modern platform for data management and analysis in big data domain R. Shree, T. Choudhury, S. C. Gupta, and P. Kumar.

The paper talks about how dealing with real-time data nowadays is quite complicated, with many technologies that need to work together. It suggests using Apache Kafka, which is like a smart system for handling streams of data. It acts as a messaging system and is good at storing data reliably. Kafka is useful for two main things: moving data between systems in real time and creating applications that work with live data streams. It works on multiple servers, storing data in categories called topics, each having a key, a value, and a timestamp. The paper explains Kafka's structure and gives examples of how it helps solve problems in the Big Data era by using streaming solutions.

6. Event-Based Sensor Data Scheduling: Trade-Off Between Communication Rate and Estimation Quality J. Wu,Q.-S. Jia, K.H. Johansson and L.Shi.

In this paper, the focus lies on addressing the challenge of sensor data scheduling for remote state estimation within networked control systems. Given the constraints of limited communication energy and bandwidth, the paper proposes an event-based sensor data scheduler tailored for linear systems. The objective is to enable sensors to make informed decisions on whether to transmit measurements to a remote estimator for further processing, striking a balance between communication rate and estimation quality. Through the derivation of a minimum mean-squared error (MMSE) estimator, the paper outlines a methodology for achieving this balance. By selecting appropriate event-triggering thresholds, the proposed scheduler aims to optimize the trade-off between sensor-to-estimator communication rate and remote estimation quality. The paper provides simulation examples to validate the proposed approach, demonstrating its effectiveness in practical scenarios. Additionally, the paper contributes to the broader field of sensor scheduling and remote estimation under communication constraints, building upon existing research while introducing novel insights and methodologies for improving system performance in resource-constrained environments.

3. System Architecture

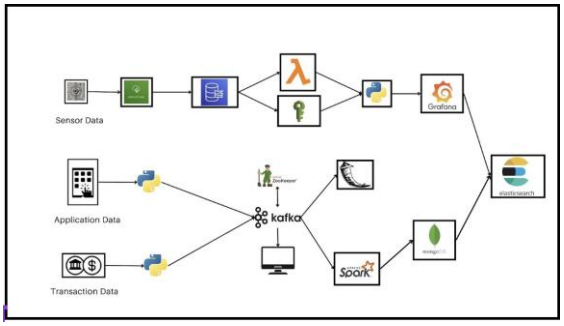

Fig 1. System Architecture.

3.1. Workflow 1 (Sensor)

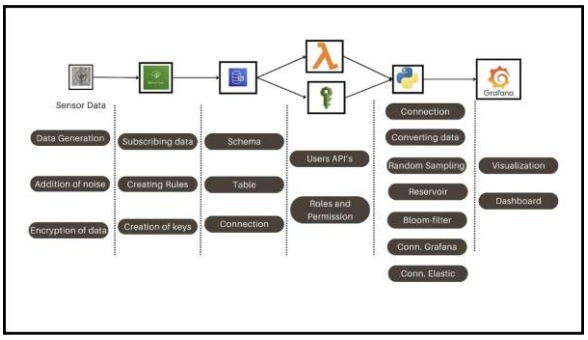

Fig 2. System Architecture for Sensors data.

In the initial phase of our project, we aimed to collect data from laptop sensors and transmit it to AWS IoT services for further analysis and visualization. To enhance the dataset's utility, we introduced synthesized or *'fake'* data alongside the original sensor data. This approach enriched the dataset, making it more representative and providing additional data points for analysis. We utilized techniques such as Bloom filters to efficiently handle large volumes of data and enable fast queries. The setup involved creating an AWS environment, setting up AWS IoT connections, and creating a database and table in AWS Timestream for schema definition. The storage time for the data was set to default, and the IAM user with appropriate permissions was created to establish a connection between Python and AWS Timestream. To interact with the data in AWS Timestream, we developed a function to fetch data in column form and decrypt it to obtain the original data. Additionally, we implemented a random sampling technique, specifically reservoir sampling, to create a sample for input into the Bloom filter. Parameters for reservoir sampling were specified, with potential tuning using different algorithms. In summary, our project workflow involved collecting sensor data, augmenting it with synthesized data, storing it in AWS Timestream, and then analyzing it using Python. Techniques such as Bloom filters and random sampling were employed to enhance data analysis and visualization capabilities, ultimately facilitating informed decision-making and driving innovation in the project.

3.2. Workflow 2 (Transaction and Application)

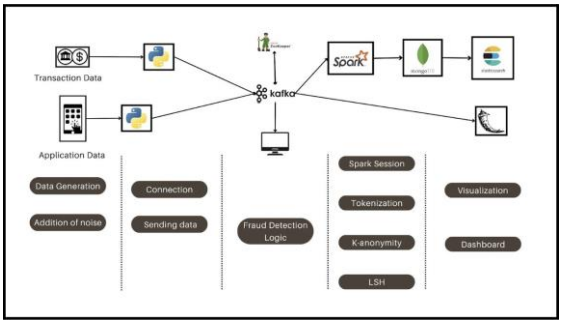

Fig 3. System Architecture for transaction and Application data .

Firstly, transaction data is collected and pushed to Kafka, a distributed streaming platform. Kafka acts as a buffer, enabling real-time data processing and facilitating data transfer between different systems. This ensures that the data is efficiently ingested and available for further processing.

Next, the data is read using PySpark, a powerful analytics engine for large-scale data processing in Python. PySpark's distributed computing capabilities allow for the efficient handling of big data sets. Once the data is loaded, a function is written to perform reservoir sampling. Reservoir sampling is a technique used to randomly select a subset of data points while preserving the overall distribution characteristics of the original data set. This sampling approach is crucial for implementing a Bloom filter with good accuracy. Data anonymization techniques are applied to protect user privacy while preserving the integrity of the data. Anonymization involves removing or encrypting personally identifiable information from the data set, ensuring that individual identities cannot be traced back to the underlying transactions.

Once the data is anonymized, Locality-Sensitive Hashing (LSH) is performed. LSH is a technique used for approximate nearest neighbor search, which helps to identify similar data points or patterns within the data set. By employing LSH, the data can be clustered based on similarity, enabling efficient fraud detection and identification of anomalous transactions.

Application data on the other hand is initially loaded into Python, where it undergoes pre-processing before being ingested into Kafka for further processing. This application data encompasses various types of information. Once ingested into Kafka, the data undergoes real-time streaming between systems, facilitating seamless communication and data flow. This is the stage where fraud logic is predicted. Subsequently, Flask is being used to generate some visuals that can help in predicting the fraud transactions based on the application data.

4. Results

Fig 4. Grafana Visualization .

The output of the sensor data is passed to Grafana and this is a real-time dashboard that has all the parameters plotted in a line graph the spikes tell us that there is a sudden change in the location of the user and that is what is not acceptable as in real world it is not possible that the use will be traveling in that such speed and so we have chosen a threshold of 0.2 to do this and this may vary depending upon the infrastructure of the institution because for us the Grafana is accepting only 3 integers and this is making more difficult to implement the logic that we thought so, it is a total waste to transport the data from python to Grafana as again it will not be seeing the digits after three decimal places. So, we have just passed the data that can be seen and easy to tell what is going on with the location.

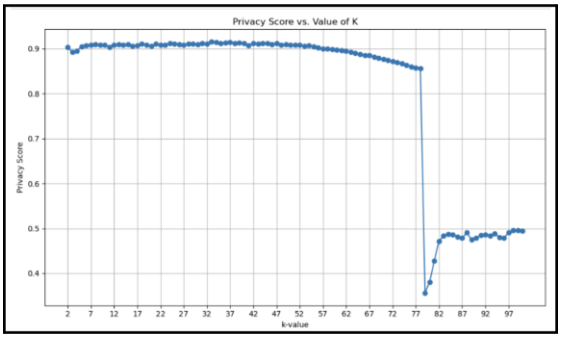

Fig 5. Best k value analysis.

The below graph is about telling the best k value suitable for the project and the best values will be selected for the data but the problem is that we have to select the optimal value for that we need a model for this step and later we will use that value to anonymize the data.

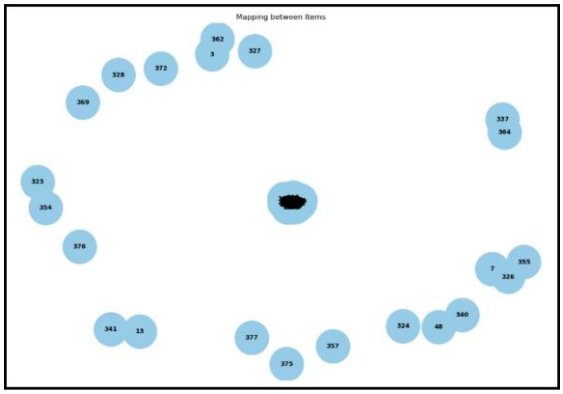

Fig 6. Cluster Representation.

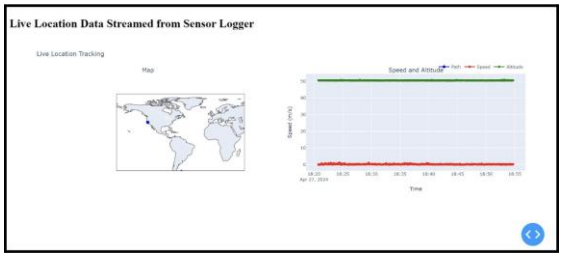

Fig 7. Live Location from Scanner Logger.

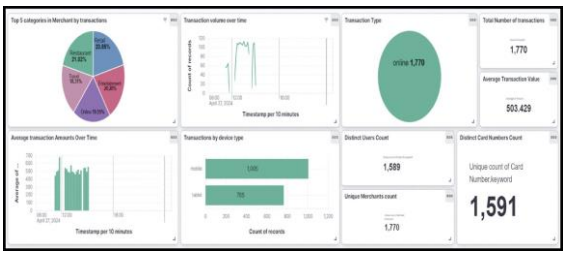

Fig 8. Dashboard from Elastic Kibana.

5. Discussion and Future work

Future works for this project could involve enhancing the fraud detection algorithms by incorporating machine learning models, such as anomaly detection algorithms or deep learning architectures, to improve the accuracy of identifying fraudulent activities. Additionally, exploring advanced data anonymization techniques, such as differential privacy or homomorphic encryption, can further bolster data security and privacy while preserving the integrity of the data. Integration of real-time monitoring and alerting systems can be implemented to provide immediate notifications of suspicious transactions or anomalies, enabling proactive fraud prevention measures. Furthermore, exploring the potential of integrating external data sources, such as social media or third-party APIs, can enrich the dataset and provide additional context for fraud detection and customer profiling. Overall, continuous optimization and refinement of the data analytics pipeline, along with staying ahead of emerging technologies and industry trends, will be crucial.

6. Conclusion

In conclusion, the implementation of a real-time data analytics pipeline tailored for the banking sector represents a significant leap forward in meeting the evolving demands of digital banking. By leveraging a robust technological stack, including Docker, Kafka, and Apache Spark, alongside innovative integrations like IOT and Timestream for sensor data, and Python for transactional and application data, this project enables efficient data processing and analysis.

The deployment of Flask, Elastic search, and Grafana further enhances the capabilities of the pipeline, empowering banking personnel with instantaneous visuals, notifications, and alerts for prompt action, particularly in critical scenarios such as fraud detection or transaction anomalies.

Through the intelligent utilization of real-time data analytics, organizations can gain invaluable insights into customer behavior, promptly identify fraudulent activities, and deliver tailored banking experiences, ultimately leading to enhanced customer satisfaction and loyalty.

This project underscores the transformative potential of real-time data analytics in revolutionizing the banking sector, emphasizing its role as a strategic shift towards a data-centric approach, poised to reshape the industry landscape and drive innovation, competitiveness, and sustainable growth in the dynamic marketplace of today and tomorrow.

References

[1] J. Wu, Q. -S. Jia, K. H. Johansson and L. Shi, ”Event-Based Sensor Data Scheduling: Trade-Off Between Communication Rate and Estimation Quality,” in IEEE Transactions on Automatic Control, vol. 58, no. 4, pp. 1041-1046, April 2013.
https://ieeexplore.ieee.org/document/6286997

[2] R. Shree, T. Choudhury, S. C. Gupta, and P. Kumar, ”KAFKA: The modern platform for data management and analysis in the big data domain,” 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), Noida, India, 2017.
https://www.semanticscholar.org/paper/KAFKA%/3A-The-modern-platform-for-data-management-and-Shree-Choudhury/90bcda14a912e1113-49f52b747f696c31adf9657

[3] F. Gürcan and M. Berigel, ”Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges,” 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2018.
https://ieeexplore.ieee.org/abstract/document/-8567061

[4] Shahrivari, S. (2014). Beyond Batch Processing:Towards Real-Time and Streaming Big Data. Computers, [online] 3(4), pp.117–129.
https://doi.org/10.3390/computers3040117

[5] Xu, L., Jiang, C., Wang, J., Yuan, J. and Ren, Y. (2014). Information Security in Big Data: Privacy and Data Mining. IEEE Access, 2(2), pp.1149–1176.
https://doi.org/10.1109/access.2014.2362522

[6] Eman Shaikh, Iman Ahmed Mohiuddin, Yasmeen Alufaisan and Irum Nahvi (2019). Apache Spark: A Big Data Processing Engine. [online] ResearchGate.
https://www.researchgate.net/publication/-339176824 /Apache /Spark /A /Big /Data_/ -Processing_/Engine.

Appendix A. Technical Difficulties

Implementing a robust real-time data analytics pipeline in the banking sector presents several technical challenges that require careful consideration and innovative solutions. One significant challenge is ensuring the scalability and fault tolerance of the infrastructure, especially when dealing with large volumes of streaming data. As the data volume increases, the system must seamlessly scale to accommodate the growing workload while maintaining high availability and reliability. This necessitates the adoption of distributed computing frameworks like Apache Spark and Kafka, which can handle data processing and streaming across multiple nodes in a fault-tolerant manner. Additionally, ensuring data quality and consistency in real-time analytics poses a considerable challenge. Real-time data streams may suffer from issues such as data duplication, out-of-order arrival, and data skew, which can impact the accuracy and reliability of analytics results. Addressing these challenges requires the implementation of data cleansing and validation mechanisms, such as schema validation, duplicate detection, and outlier detection, to ensure that only high-quality data is processed and analyzed. Furthermore, ensuring data privacy and security is paramount in the banking sector, where sensitive financial information is involved. Implementing robust encryption, access control, and data anonymization techniques is essential to protect customer data and comply with regulatory requirements such as GDPR and CCPA. Moreover, integrating disparate data sources and systems within the banking environment presents interoperability challenges. Banks typically operate with a diverse range of legacy systems, databases, and third-party applications, each with its own data formats, protocols, and APIs. Achieving seamless data integration and interoperability requires the development of custom connectors, data adapters, and middleware layers to facilitate data exchange and communication between disparate systems.Overall, addressing these technical challenges requires a holistic approach that combines expertise in distributed systems, data engineering, cybersecurity, and regulatory compliance to ensure the successful implementation of real-time data analytics solutions in the banking sector.

Appendix B. Innovation

Innovations in the realm of real-time data analytics for the banking sector represent a pivotal avenue for advancing operational efficiency and customer satisfaction. Building upon established frameworks such as Apache Spark and Kafka, novel approaches are being developed to address pressing challenges and unlock new opportunities. Advanced fraud detection models harness the power of real-time streaming data to identify fraudulent activities swiftly and accurately, safeguarding both the bank and its customers. Dynamic customer segmentation models leverage real-time data to tailor services and marketing strategies with precision, ensuring personalized experiences for each customer. Predictive maintenance initiatives integrate sensor data and predictive analytics to preemptively address potential system failures, minimizing downtime and enhancing reliability. Real-time personalized recommendation engines utilize streaming data to offer tailored product and service suggestions, enriching customer interactions and fostering loyalty. Continuous compliance monitoring systems analyze real-time data streams to ensure regulatory adherence and detect anomalies promptly, mitigating compliance risks. Blockchain technology is explored to protect the integrity and security of banking data, leveraging its decentralized ledger for real-time transparency and trust. Additionally, sentiment analysis models tap into real-time social media and news data to gauge public perception, empowering banks to proactively manage their brand reputation. Lastly, the potential of quantum computing in revolutionizing real-time analytics is investigated, promising faster and more efficient processing of data streams for enhanced decision-making and risk management. These innovative efforts underscore the transformative potential of real-time data analytics in reshaping the banking landscape, driving innovation, and delivering unparalleled value to customers and stakeholders alike.

Appendix C. Lesson Learnt

The exploration of real-time data analytics within the banking sector has yielded invaluable lessons

showing the importance of technological innovation and strategic implementation. Firstly, the adoption of a comprehensive technological stack, including Docker, Kafka, and Apache Spark, has demonstrated the significance of selecting robust tools capable of handling complex data streams efficiently. This highlights the necessity of investing in cutting-edge technologies to ensure seamless data processing and analysis. Secondly, the integration of Change Data Capture (CDC) techniques has proven pivotal in capturing and analyzing real-time data alterations, emphasizing the importance of staying abreast of emerging methodologies to enhance data capture capabilities. Thirdly, the implementation of advanced fraud detection models and dynamic customer segmentation strategies has showcased the transformative potential of real-time analytics in mitigating risks and enhancing customer experiences. This underscores the importance of leveraging data-driven insights to drive innovation and strategic decision-making within the banking sector. Additionally, the exploration of emerging technologies such as blockchain and quantum computing has highlighted the need for continuous experimentation and adaptation to harness the full potential of evolving technologies. Overall, the project has emphasized the critical role of real-time data analytics in revolutionizing banking operations and delivering tailored, responsive services to customers, thereby paving the way for future advancements in the field.

Appendix D. Prospects of winning competition/ Publication

The real-time data analytics project in the banking sector represents traditional banking operations - through cutting-edge technological advancements. By utilizing the power of Change Data Capture (CDC) techniques, Apache Spark, Kafka, and Docker, the project ensures seamless data streaming, processing, and analysis in real-time. This robust technological stack enables swift detection of fraudulent activities, dynamic customer segmentation, and personalized banking experiences, thereby addressing critical issues such as fraud detection and enhancing customer trust and satisfaction. Moreover, the project's emphasis on data privacy and security measures, including anonymization techniques and encryption protocols, not only ensures regulatory compliance but also sets a benchmark for responsible data management practices across industries. Beyond banking, the project's innovative methodologies and scalable infrastructure hold transformative potential for various sectors, including e-commerce, healthcare, and supply chain management. By offering real-time insights, proactive risk management, and enhanced customer experiences, the project paves the way for cross-industry adoption of real-time data analytics, driving operational efficiencies and fostering innovation in an increasingly data-driven world.

Appendix E. Pair Programming

Pair programming can significantly enhance the efficiency and effectiveness of various stages and components within the real-time data analytics project for the banking sector. Firstly, in the initial phase of data preprocessing and analysis, we collaborated to ensure the cleanliness and consistency of raw data collected from banking systems. Together, we employed advanced techniques such as feature engineering and anomaly detection using tools like PySpark and Pandas. Secondly, in the domain of integration and testing, paired up as teams to seamlessly integrate various components of the data pipeline, including Kafka streams, Apache Spark jobs, and Elasticsearch indices. Through collaborative efforts, as a result we were able to comprehensive unit tests, integration tests, and end-to-end tests to validate the correctness and reliability of the entire system. Thirdly, in dashboard development, implemented interactive dashboards using tools like Grafana and Flask. Focused on real-time data streaming integration to deliver intuitive and insightful dashboards. Furthermore, in infrastructure setup and deployment, to configure and deploy the necessary infrastructure on cloud platforms like AWS. Collaborating on resource provisioning, networking configuration, and Docker container deployment, ensured scalability, reliability, and security of the system. Lastly, in documentation and knowledge sharing, documenting design decisions, implementation details, and best practices. Overall, by implementing pair programming in these critical areas, we were able to achieve collective expertise, foster collaboration and communication, and accelerate the development and deployment of the real-time data analytics solution for the banking sector.