Capstone Project

AML Transaction Monitoring System

Using Apache Kafka and Machine Learning for fraud detection

Documented by Jennie Tang

Date: 14 May 2025

Table of Content

1. Problem and Opportunity

Financial institutions face significant challenges in detecting money laundering activities in real-time. Traditional batch processing methods introduce substantial delays between suspicious transactions and their detection, creating windows of opportunity for financial criminals (TrustDecision, 2024).

According to former U.S. Deputy Attorney General Paul McNulty, “If you think compliance is expensive – try non-compliance.” An estimated 2-5% of global GDP ($800 billion to $2 trillion) is laundered annually, yet less than 1% of illicit financial flows are seized by authorities (United Nations Office on Drugs and Crime, 2024).

To address this gap, effective anti-money laundering (AML) software provides real-time risk assessments by evaluating transactions flows and flagging abnormalities using advanced algorithms. This allows compliance teams to respond promptly to potential threats.

This project proposes a real-time AML monitoring solution that combines Apache Kafka’s streaming capabilities with machine learning-driven risk scoring. By detecting suspicious activity as it happens, rather than hours or days later, we aim to close the detection-response gap and enhance financial system integrity.

Current State	Desired State
Batch processing with 24+ hour delays	Real-time transaction screening
High false positive rates (90-95%)	F1-optimized ML models with significantly lower false positives
Manual review of most alerts	Automated risk scoring with human review only for high-risk cases
Strict rule-based systems	Adaptive, learning systems that improve over time

Table 1: Transition from traditional AML practices to intelligent real-time monitoring

Prior approaches to AML detection have primarily relied on rule-based systems or basic statistical methods. Recent research has begun exploring machine learning for AML, but few solutions have successfully combined streaming architecture with advanced ML algorithms in production environments. Notable research includes:

- Weber et al. (2019) achieved 89% reduction in false positives using deep learning models for AML but relied on batch processing.
- The FINCEN Files investigation (2020) revealed the limitations of current systems, showing vast sums moving through the financial systems despite existing AML controls.
- The PaySim simulator provided a foundation for generating synthetic transaction data that mimics real-world fraud patterns.

This project builds upon these foundations while addressing their limitations – namely, delayed processing and limited scalability through an integrated, real-time streaming framework.

2. Industry/Domain

This project is situated at the intersection of financial services and regulatory technology, specifically focusing on AML compliance. The AML domain encompasses the policies, procedures, laws, and technologies designed to prevent criminals from disguising illegally obtained funds as legitimate income.

Figure 1: Hierarchy of Money Laundering (Source: AML Watcher)

The current finance compliance industry is undergoing significant transformation due to several factors:

- **Regulatory pressure**
  Global AML regulations continue to tighten, with significant penalties for non-compliance. The EU's 6th Anti-Money Laundering Directive, the US Anti-Money Laundering Act of 2020, and similar regulations globally are increasing requirements for financial institutions.
- **Technological disruption**
  Traditional financial institutions face challenges from FinTech start-ups offering faster, more customer-friendly services while still needing to maintain compliance. Legacy systems struggle to adapt to new requirements (Euvic, n.d.).
- **Rising compliance cost**
  Financial institutions spend an estimated $213.9 billion annually on financial crime compliance worldwide, with AML compliance representing a significant portion of these costs (LexisNexis, 2021).
- **Evolving criminal techniques**
  Money launderers continuously adapt their methods to evade detection, requiring more sophisticated monitoring techniques.

3. Stakeholders

The key stakeholders this project aims to address are…

Stakeholder	Description	Expectations
Compliance officers*	**Professionals responsible for ensuring adherence to AML regulations**	**Reduced false positives, more efficient investigations, defensible compliance processes**
Data science/ ML teams	Specialists responsible for model development and improvement	Accessible data pipelines, ability to update models, clear performance metrics
IT operations teams	Staff responsible for maintaining technical infrastructure	Reliable system performance, manageable maintenance requirements, clear monitoring capabilities
Risk management teams*	**Departments assessing and mitigating various organizational risks**	**Better risk visibility, quantifiable risk metrics, improved risk controls**
Financial institution executives*	**C-suite leaders responsible for overall compliance strategy**	**Reduced compliance costs, minimized regulatory risk, protection of institutional reputation**
Financial intelligence units (FIUs)	Government agencies responsible for combating money laundering	Higher quality suspicious activity reports, improved intelligence for investigations
Banking customers	End users of financial services	Minimal transaction friction, protection from being unwittingly involved in financial crime
Core users are the primary target for system utility		

Table 2: Key stakeholders to AML transaction monitoring system

3.1. Core Users – Operational Decision-Makers

Core users represent the primary audience for the AML monitoring system's outputs. *Compliance Officers* are on the front lines of regulatory enforcement and rely on timely alerts to initiate investigations, reduce false positives, and ensure that Suspicious Transaction Reports (STRs) are defensible.

Risk Management Teams, while more broadly focused on organizational risk, benefit from quantifiable metrics and risk visibility to enhance internal controls and reporting.

Meanwhile, *Financial Institution Executives*, including roles like the Chief Risk Officer (CRO) and Chief Compliance Officer (CC), oversee the strategic direction of compliance

infrastructure. They are especially concerned with minimizing regulatory penalties, reducing compliance costs, and protecting the institution's reputation.

This group's engagement is critical to ensure that the system is usable, interpretable, and aligned with compliance objectives.

3.2. Technical Stakeholders – Developers and Maintainers

This group comprises of *Data Science and ML Teams* who are responsible for developing, validating and refining the predictive models that assess transaction risk. Their tasks include ensuring that the models are optimized for real-time use cases. They are also responsible for continuous model updates based on new fraud patterns.

IT Operations Teams support the technical infrastructure, including Apache Kafka, Gradio and the real-time pipeline. They are responsible for scaling data ingestion pipelines, and maintaining secure access control.

These teams are essential for maintaining system performance and minimizing technical debt. Without close coordination between ML and IT operations, the system risks becoming either a bottleneck or a compliance liability.

3.3. Non-Technical Stakeholders – Indirect Users

Although not directly interfacing with the system, non-technical stakeholders are significantly impacted by its outcomes. *Financial Intelligence Units (FIUs)*, referring to government bodies like Suspicious Transaction Reporting Office (STRO SG), consume STRs and intelligence reports generated downstream of AML systems. These agencies benefit from high-quality, timely and actionable intelligence to pursue investigations and disrupt criminal networks.

Banking Customers, the end-users of financial services, are perhaps the most non-involving in the system, yet they are heavily impacted by its success or failure. Effective AML monitoring protects them from becoming unwittingly participants in financial crimes and ensure minimal friction during legitimate transactions. Their trust in the financial institution hinges partly on invisible but effective compliance infrastructure.

4. Business Question

A business question focuses on organizational goals, outcomes, or decisions, and is asked by strategic stakeholders like executives and compliance officers. *Figure 2* presents the central business question driving this project.

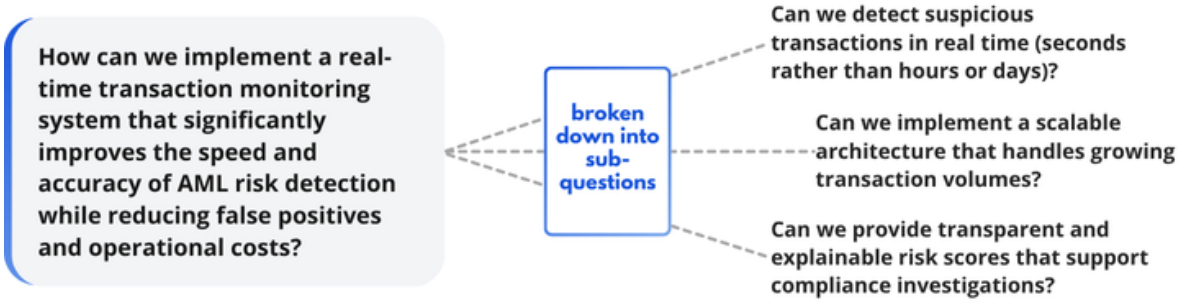

Figure 2: Business question

The business value of an effective real-time AML system can be quantified across several dimensions.

Value Dimension	Conservative Estimate	Assumptions
Reduction in false positives	$2-5 million annually	Based on 40-60% reduction in false positives x average investigation cost of $25-50 per alert x 200,000 alerts annually.
Improved regulatory standing	$1-10 million annually	Reduced probability of regulatory actions and penalties by 10-20%.
Operational efficiency	$0.5-2 million annually	Streamlined investigation workflows, reduced manual review time, more efficient resource allocation.
Enhanced risk detection	Not directly quantifiable	Improved ability to detect actual money laundering provides significant protection against financial and reputational damage.

Table 3: Business value to answering business question

For a mid-to-large financial institution, the total annual value is conservatively estimated at $4.5-20 million, with an expected ROI of 200-500% over three years on the technology investment.

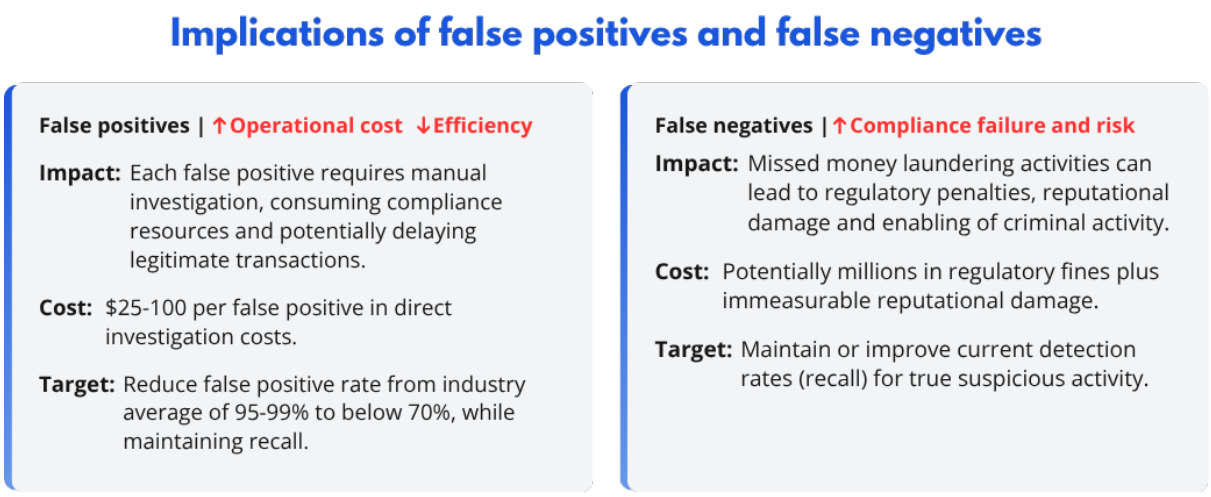

Figure 3: Implications of false positives and false negatives

Given the asymmetric nature of these costs, the system incrementally improves precision to reduce false positives, and prioritizes maintaining high recall to minimize false negatives. The target performance would be to maintain or improve recall while achieving at least a 30% reduction in false positives compared to current systems. This also addresses the above-mentioned desired state *(See Table 2: Transition from traditional AML practices to intelligent real-time monitoring)*.

5. Data Question

On the other hand, data question focuses on what needs to be measured, modelled, or analyzed, and is asked by data scientists, ML engineers and analysts. A business question drives the “why”; while a data question defines the “how”. *Figure 4* highlights the overarching data question guiding the technical approach in this project.

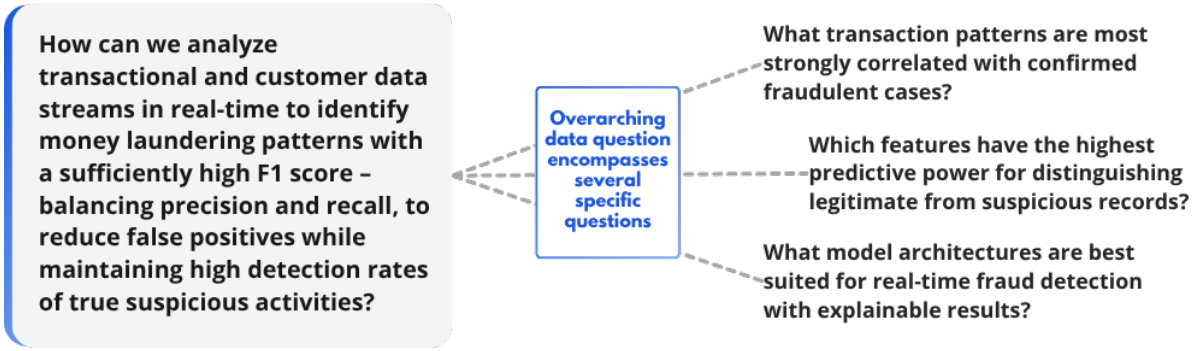

Figure 4: Data question

To build an effective real-time AML monitoring system, we need several types of data. See *Table 4 – Data requirement*.

Data Category	Description/ Purpose
Transaction data	Records of financial transactions including amount, time, transaction type, origin and destination that simulates both legitimate and suspicious patterns.
Labelled cases	Previously confirmed fraud cases and false positive cases for training and evaluating machine learning models.

Table 4: Data requirement

6. Data

6.1. Data Source

The primary dataset used in this project is the PaySim dataset, available from Kaggle at https://www.kaggle.com/datasets/ealaxi/paysim1/data. PaySim is a synthetic financial dataset for fraud detection that simulates mobile money transactions and injects fraudulent behaviour. The original logs were provided by a multinational company that operates mobile financial services in more than 14 countries globally. This synthetic dataset is scaled down to a quarter of the original dataset’s size.

6.2. Volume and Attributes of the Data

The PaySim dataset contains approximately 6.3 million transaction records with 11 features.

Feature	Description	Data Type
step	Hour of simulation (1 step = 1 hour)	Integer (1-744)
type	Transaction type (CASH-IN, CASH-OUT, DEBIT, PAYMENT, TRANSFER)	Categorical
amount	Transaction amount	Float
nameOrig	Customer initiating the transaction	String
oldbalanceOrg	Initial balance of originator	Float
newbalanceOrig	New balance of originator after transaction	Float
nameDest	Customer receiving the transaction	String
oldbalanceDest	Initial balance of recipient	Float
newbalanceDest	New balance of recipient after transaction	Float
isFraud	Fraud indicator (target variable)	Binary (0/1)
isFlaggedFraud	Flag for transactions over 200,000 in amount	Binary (0/1)

Table 5: Attributes of PaySim dataset

Initial data quality assessment reveals there is no missing value in the dataset. Data types are consistent across all columns with synthetic customer IDs that excluded personally identifiable information.

6.3. Reliability of Data

As a synthetic dataset, PaySim presents both strengths and limitations. It offers several advantages that make it suitable for research and prototyping. First, it eliminates privacy concerns by excluding any personally identifiable information. Second, transaction records are simulated based on patterns observed in real financial data, which ensures we draw realistic insights from the data. Third, all transactions are comprehensively labelled as either fraudulent or legitimate, facilitating supervised learning tasks. Most importantly, PaySim is well-documented and widely adopted in academic research, lending credibility and reproducibility to studies that use it.

Despite its advantages, the PaySim dataset has several limitations that must be considered. It provides a simplified representation of the complexities inherent in real-world financial systems, which may not fully reflect the nuanced behaviour of diverse customer segments or transaction types. Notably, the dataset is constrained to specific fraud patterns that were pre-programmed into the simulator, potentially overlooking less common or more sophisticated techniques used in modern money laundering schemes. As such, it does not capture the full diversity or evolving nature of financial crime.

To ensure models trained on this dataset can generalize effectively to real-world application, subsequent validation and fine-tuning using authentic transactional data would be a valuable step in future development.

6.4. Data Availability and Real-Time Simulation Strategy

The PaySim dataset is a static synthetic dataset designed to simulate mobile money transactions based on historical patterns. It does not receive real-time updates or reflect live transaction flows. To address this limitation during development and testing, this project incorporates a custom transaction generator, which programmatically simulates continuous transaction streams derived from the structure and statistical characteristics of PaySim. This enables the system to mimic real-time data ingestion and apply machine learning inference in a streaming context using Apache Kafka.

7. Data Science Process

This section outlines the complete data science process implemented in this project, from initial data analysis through model development to real-time implementation on Gradio dashboard with Apache Kafka. A visual summary of the full deployment pipeline is provided in *Figure 16*.

7.1. Data Analysis

The initial data exploration revealed several key insights.

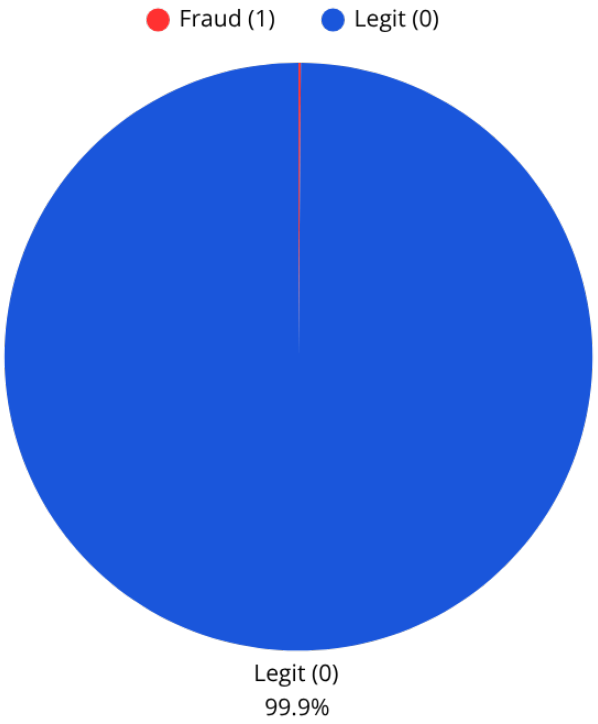

Figure 5: Class distribution

Figure 5 shows significant class imbalance was observed with only 0.129% of transactions are fraudulent.

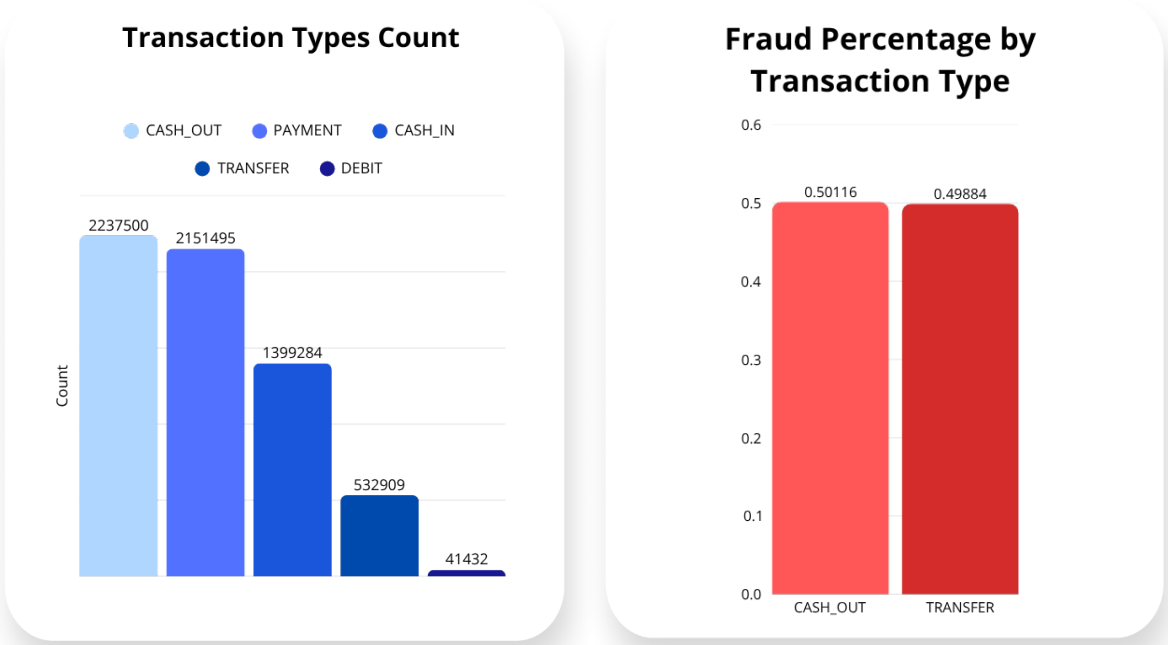

Figure 6: Transaction type and fraud distribution

Figure 6 illustrates transaction type and fraud distribution across the dataset. In the *left plot*, transaction volumes are highest for CASH_OUT, followed closely by PAYMENT, then CASH_IN and TRANSFER, with DEBIT transactions being the least frequent. The *right plot* highlights the concentration of fraudulent activity, showing that all identified fraud cases occur exclusively within the TRANSFER and CASH_OUT transaction types.

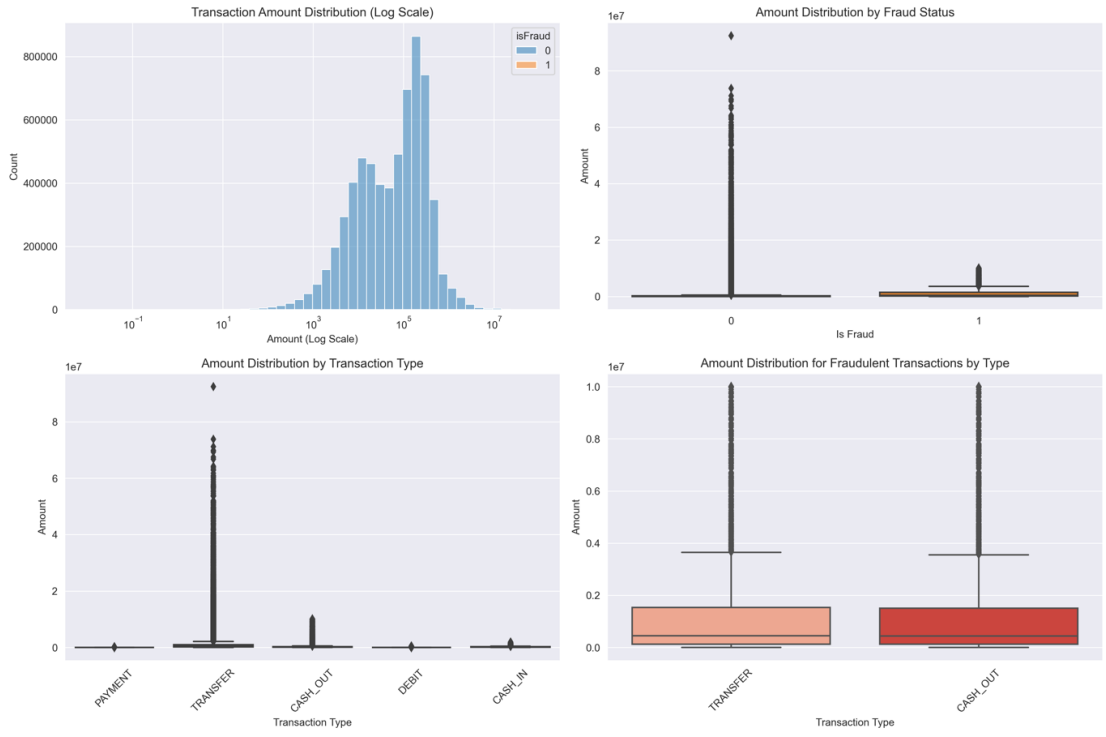

Figure 7: amount distribution analysis

Figure 7 presents various views of transaction amount distribution.

The *top-left histogram* shows a heavily right-skewed distribution, with most transactions ranging between $1,000 to $100,000. Fraudulent transactions are less visible due to their rarity.

The *top-right boxplot* compares amounts by fraud status, revealing that non-fraud transactions include higher-value outliers, while fraud tends to cluster in the mid-to-high range, suggesting subtler exploitation patterns.

The *bottom-left plot* shows that TRANSFER transactions have the widest range and highest outliers, highlighting their potential risk exposure.

Finally, the *bottom-right plot* confirms that fraud occurs only in TRANSFER and CASH_OUT types, both showing relatively tight distributions with median value below $2 million, implying potential thresholds that fraudsters may repeatedly target.

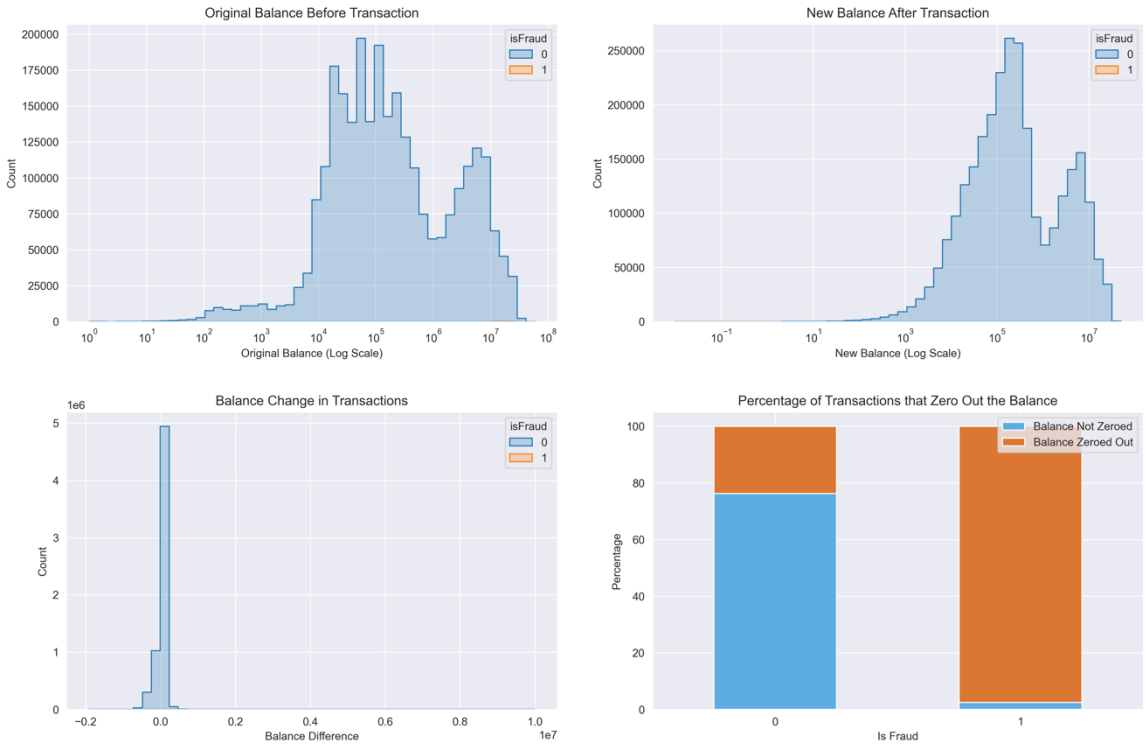

Figure 8: Balance analysis

Figure 8 explores balance behaviours before and after transactions. A new feature deltaOrig was created for this analysis which reflects the amount debited from the sender's account during the transaction. The most prominent plot here is in the bottom-right, which suggests fraudsters often withdraw significant sums – the rightmost bar chart shows that a large proportion of fraudulent transactions completely deplete the origin account, reinforcing the idea of rule-based or scripted fraud behaviour.

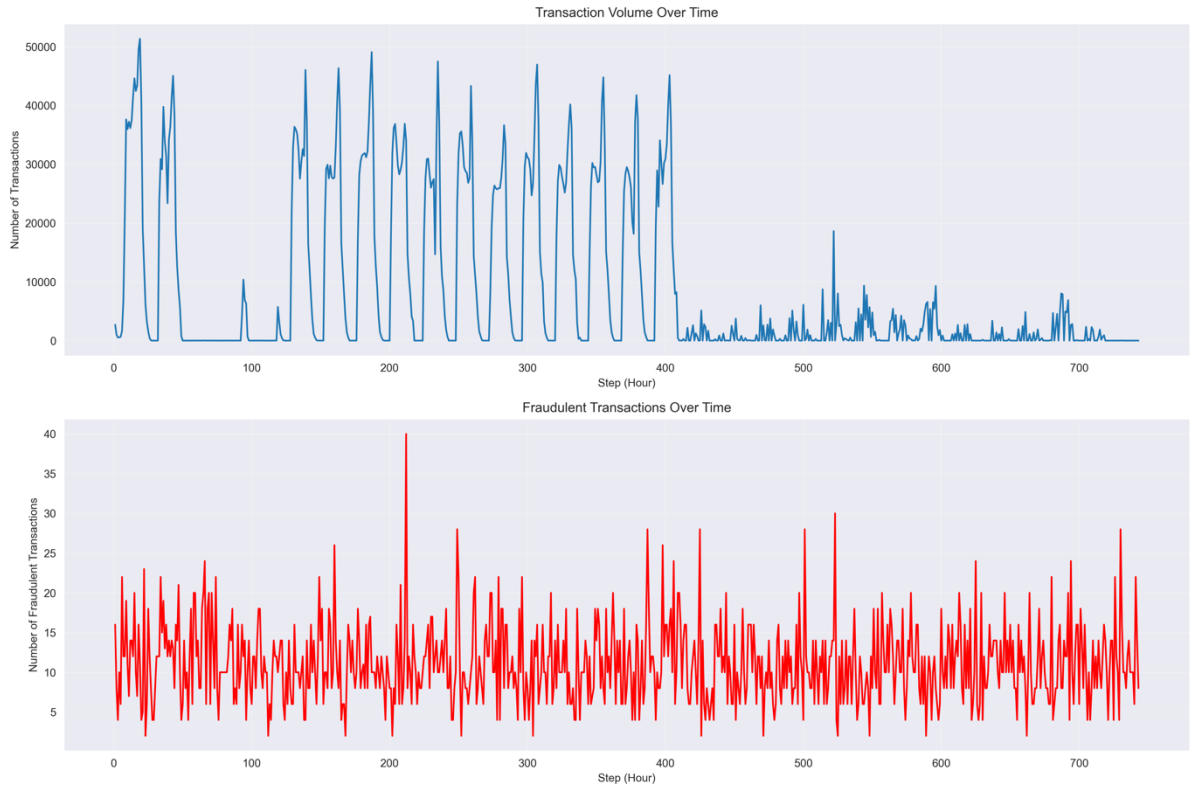

Figure 9: Temporal analysis

Figure 9 exhibits transaction volume and fraud volume over time. The *top chart* shows a clear periodicity with repeated sharp spikes and valleys up to around step 400. From that point onwards, volume drops significantly and becomes erratic. The *bottom chart* has shown that fraud activity remains relatively stable, ranging between 5 – 25 cases per hour despite a drop in total transaction volume from step 400. This confirms that fraud volume is weakly correlated with overall transaction volume, suggesting that fraud occurs consistently as if the transactions were run on script or automated to avoid detection through volume anomalies.

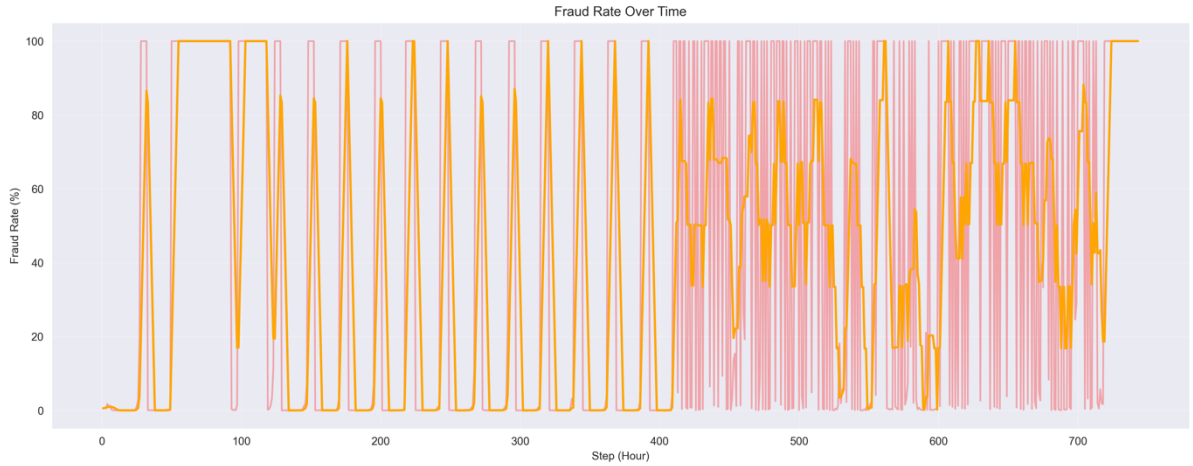

Figure 10: Fraud rate over time

Figure 10 reveals a noticeable shift in fraud patterns after step 400, where the fraud rate becomes increasingly volatile, exhibiting recurring spikes every 20 to 30 hours. Interestingly, these peaks tend to occur during low-volume transaction period, which may indicate that fraudsters are deliberately timing their activities to avoid detection. This pattern suggests a prolonged or staged attack strategy, potentially aimed at probing the system's fraud detection responsiveness during quieter periods.

7.2. Feature Engineering

Several derived features were created to improve model performance.

Engineered Features	Code snippet	Rationale
oridAccountType, destAccountType	`data['nameOrig'].str[0]` `data['nameDest'].str[0]`	Extracts first character to identify account type (e.g., C = customer, M = merchant)
C_to_C	`((df['origAccountType'] == 'C') & (df['destAccountType'] == 'C')).astype(int)`	Flags customer-to-customer transfers, where fraud is more prevalent
logAmount	`np.log1p(data['amount']`	Reduces skewness and stabilizes variance for high-value transaction amounts
oldbalanceDiff, newbalanceDiff	`data['oldbalanceOrg'] – data['oldbalanceDest']…`	Captures imbalance patterns between sender and receiver before/after transfer
balanceChangeOrig, balanceChangeDest	`Data['newbalanceOrig']– data['oldbalanceOrg']…`	Quantifies how much the balances changed due to the transaction
isOriginZeroed	`(data['oldbalanceOrg'] > 0) & (data['newbalanceOrig'] == 0)`	Flags cases where the transaction fully depletes the sender's balance
amountToOldBalanceRatio	`data['amount'] / (data['oldbalanceOrg'] + epsilon)`	Detects if the transaction consumes a large portion of the sender's balance
isAmountCloseToBalance	`( abs(data['amount'] – data['oldbalanceOrg']) / (data['oldbalanceOrg'] + epsilon) < 0.05 ) & (data['oldbalanceOrg'] > 0)`	Identifies transactions suspiciously close to the full account balance
errorBalanceOrig, errorBalanceDest	`~np.isclose(data['newbalanceOrig'] + data['amount'], data['oldbalanceOrg']) …`	Flags inconsistencies in balance updates — a possible sign of data tampering
One-hot encoded features	`pd.get_dummies(df, columns=['type', 'origAccountType', 'destAccountType'], drop_first=True)`	Encodes categorical variables for model compatibility

Table 6: Engineered features

7.3. Modelling Approach

Feature selection

Feature selection was guided by a combination of exploratory data analysis (EDA), domain intuition and model interpretability tools. Patterns uncovered during EDA, such as frequent full balance wipe-outs and fraud being exclusive to TRANSFER and CASH_OUT transaction types directly informed the creation of engineered features like isOriginZeroed, C_to_C, and isAmountCloseToBalance. Statistical summaries, visualizations and correlation analysis were used to assess each variable's relationship with fraud occurrence. Following model training, feature importance scores were evaluated to validate initial assumptions and refine the input set based on their contribution to predictive performance.

In the second iteration of model development, several columns were removed — including balanceChangeDest, isFlaggedFraud, nameDest, nameOrig, and step. These features were either found to have low predictive value or were deemed irrelevant to the underlying mechanics of fraud. In contrast, the newly engineered features were incorporated as input variables, with isAmountCloseToBalance emerging as the most important feature *(See Figure 11)*. This finding reinforces insights from the earlier exploratory data analysis (EDA), where many fraudulent transactions were observed to completely deplete the origin account balance, suggesting a common behaviour pattern among fraudsters.

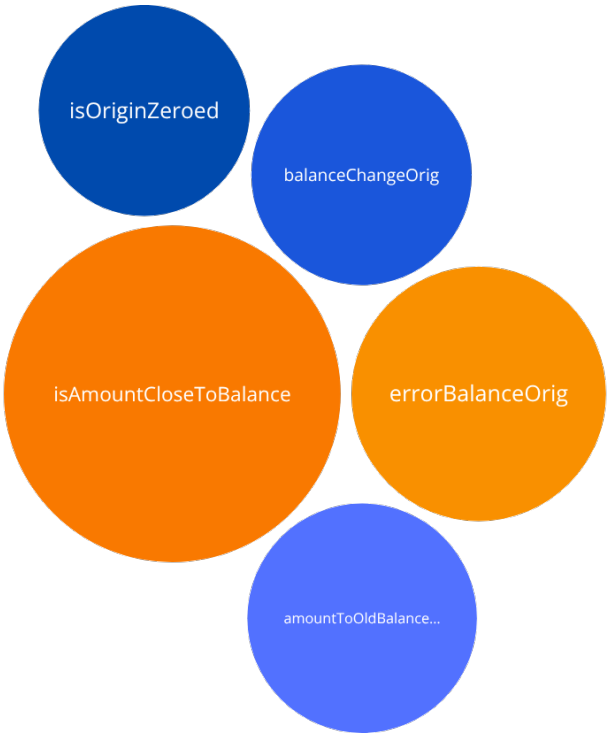

Figure 11: Top 5 important features

Model selection and outcome

Four classification models were evaluated for this fraud detection task: Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, and XGBoost Classifier. Each model was wrapped in a scikit-learn pipeline with standardized scaling and relevant class balancing techniques to address the dataset's inherent class imbalance. Among the models tested, the Random Forest Classifier consistently delivered the highest overall performance, achieving

an F1-score of 0.92, Precision of 0.85 and ROC AUC of 0.99 *(Figure 12)*. This model struck the best balance between recall and precision, making it the most suitable choice for minimizing false positives while maintaining strong fraud detection capability.

	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	RandomForestClassifier	0.999774	0.852025	0.998783	0.919585	0.999355
1	XGBClassifier	0.999574	0.752407	0.998783	0.858264	0.999631
2	GradientBoostingClassifier	0.999062	0.579449	0.998783	0.733408	0.998411
3	LogisticRegression	0.996787	0.286388	0.997565	0.445018	0.999399

Figure 12: Model evaluation

Consequently, the Random Forest model was selected for deployment in the real-time inference pipeline. As shown in *Table 3*, implementing this model could yield an estimated $2-5 million in annual savings through a 40- 60% reduction in false positives – a direct outcome of the improved precision achieved by our selected model.

Model deployment and implementation

Model deployment was carried out by exporting the best-performing model from the training pipeline using joblib, following thorough evaluation and fine-tuning. The trained model, along with the preprocessing steps were encapsulated in a full pipeline and packaged for real-time inference. This pipeline was then integrated into the Kafka consumer logic, allowing streaming transactions ingested via Kafka producer to be scored live. The entire system was containerized using Docker for portability and deployed onto a Gradio dashboard, which provides real-time monitoring of flagged, high-risk transactions, as illustrated in the visual summary in *Figure 16*. This simple web interface serves as a lightweight web interface *(Figure 13)* for demonstration purposes, acting as a stepping stone toward a fully production-ready monitoring system.

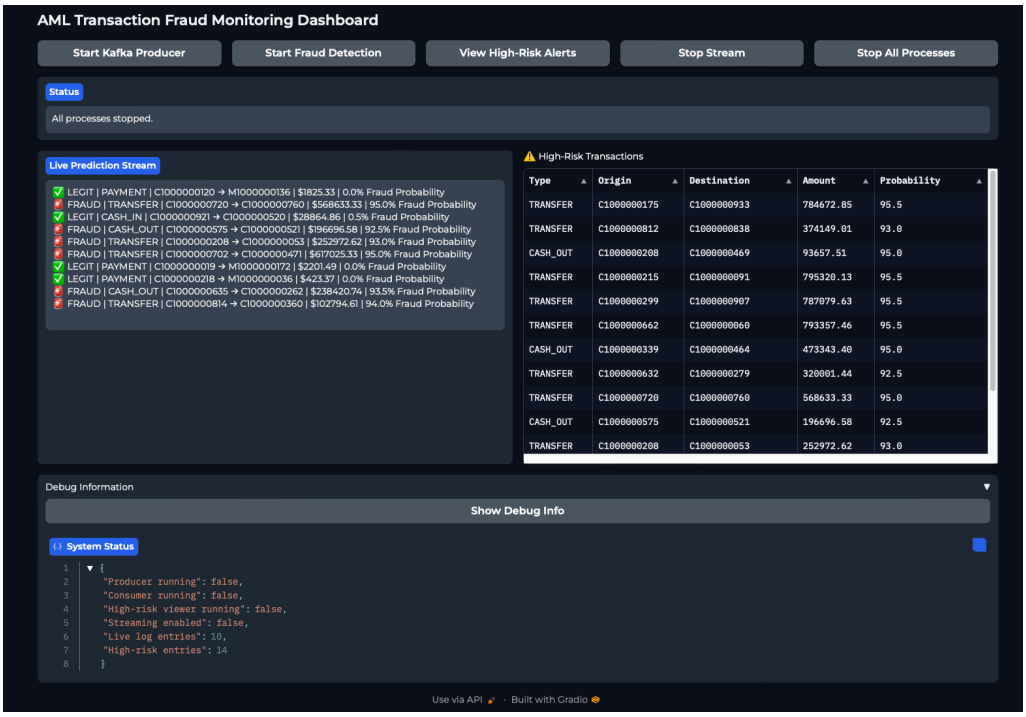

Figure 13: AML monitoring dashboard demo

8. Data Answer

The primary data question was: “How can we analyze transactional and customer data streams in real-time to identify money laundering patterns with a sufficiently high F1-score to reduce false positives while maintaining high detection rates of true suspicious activities?” This question was satisfactorily answered through our data science process – *see Figure 16.*

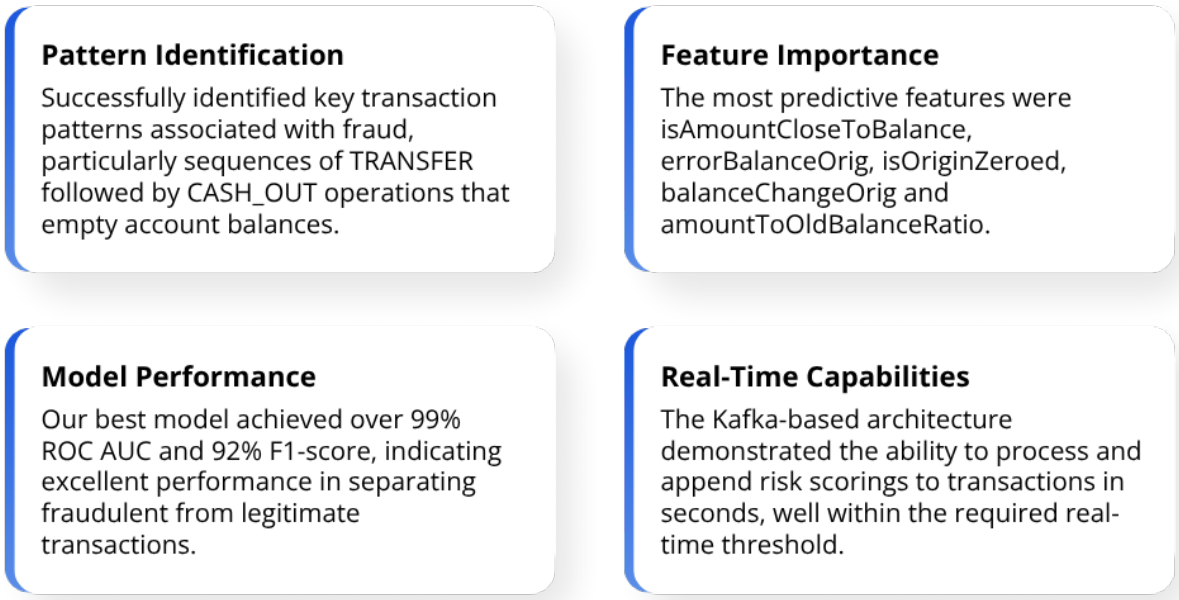

Figure 14: Data answer key insights

The confidence level in the data answer is high, with some considerations – *see Table 7.*

Aspect	Confidence Level	Rationale
Model performance	Very high	Models consistently achieved above 99% AUC
Feature engineering	High	Derived features shown strong predictive power and align well with domain knowledge
Real-time processing	High	Kafka infrastructure demonstrated sub-second processing times
Generalizability to real-world data	Medium	Performance on synthetic data may not fully transfer to real-world scenarios with evolving fraud patterns
Adaptability to new patterns	Medium	Current models may need regular retraining to adapt to new fraud techniques or patterns

Table 7: Confidence level in data answer

While PaySim is based on real transaction patterns, real-world money laundering schemes may be more complex and adaptive than those represented in the dataset. Nevertheless, the methodology and architecture demonstrated in this project are sound and can be adapted as more diverse and complex data becomes available.

9. Business Answer

The primary business question was: “How can we implement a real-time transaction monitoring system that significantly improves the speed and accuracy of AML risk detection while reducing false positives and operational costs?” This question has been satisfactorily answered through this project, which has demonstrated:

- **Real-time detection**
  A Kafka-based architecture capable of processing and scoring transactions in sub-second timeframes.
- **Reduced false positives**
  F1-focused modelling techniques that reduce false positives while maintaining high recall for actual fraud.
- **Cost efficiency**
  An architecture that can be deployed and scaled cost-effectively using modern cloud infrastructure.
- **Operational integration**
  A complete system that integrates with existing transaction flows via Kafka and provides user interfaces via Gradio.

These outcomes directly align with AML Watcher’s observation that “an efficient and upgraded version of the anti-money laundering software solutions provides real-time transaction monitoring and informs compliance experts to take immediate steps to reduce the possible damage.” (AML Watcher, 2024).

The confidence level in the business answer is high, with supporting considerations detailed in *Figure 15.* As AML Watcher (2024) aptly notes: *“Compliance is not a one-time event but rather an ongoing process.”* Our architecture supports this ongoing process by enabling efficient model updates and providing a foundation for continuous monitoring and improvement.

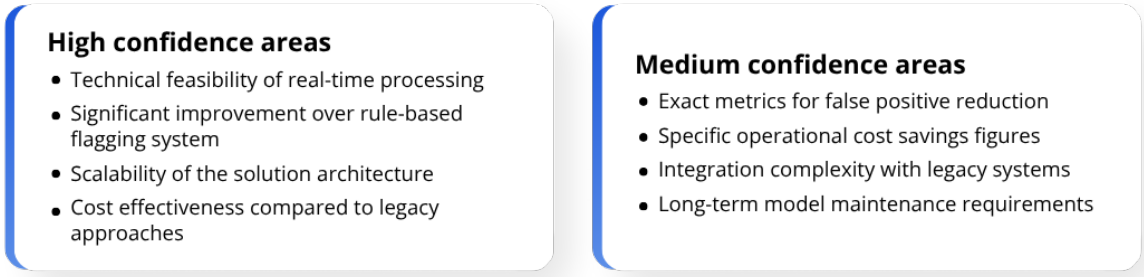

Figure 15: Confidence level in business answer

10. Response to Stakeholders

Key messages for Different Stakeholders

Stakeholder	Key Messages	Recommendations
Compliance officers	• 40-60% reduction in false positives • Real-time risk scoring of transactions • More time to focus on genuine suspicious activities	• Participate in model training and tuning • Help define risk thresholds • Provide feedback on alert quality
Data science/ ML teams	• Model achieves 99% AUC on test data • Framework for continuous improvement • Feature importance insights	• Establish regular model retraining schedule • Expand feature engineering efforts • Develop A/B testing framework for improvements
IT operations teams	• Scalable, container-based architecture • Integration via standard Kafka interfaces • Monitoring and logging built-in	• Plan infrastructure requirements • Develop operational procedures • Establish monitoring and alerting systems
Risk management teams	• Enhanced visibility into transaction risks • Quantifiable metrics on risk exposure	• Incorporate system outputs into risk reporting • Develop processes for risk threshold adjustments • Monitor effectiveness of controls
Financial institution executives	• System can provide estimated annual savings of $4.5-20M • Enhanced regulatory compliance posture • ROI of 200-500% over three years	• Approved phased implementation • Allocate resources for implementation team • Establish clear metrics for success

Table 8: Key messages and recommendations to stakeholders

11. End-to-End Solution

Our end-to-end architecture is built on a streaming data processing model. To simulate live transaction flow for demonstration, a custom transaction generator feeds data into designated Kafka topics, replicating input from upstream systems. These transactions are consumed by a fraud detection service that assigns real-time risk scores. Transactions with fraud probability exceeding 30% are classified as high-risk and routed to a separate Kafka topic for alerting purposes. In addition, all scored transactions are logged, with high-risk cases specifically written to a CSV file, enabling compliance officers to quickly review and take appropriate action.

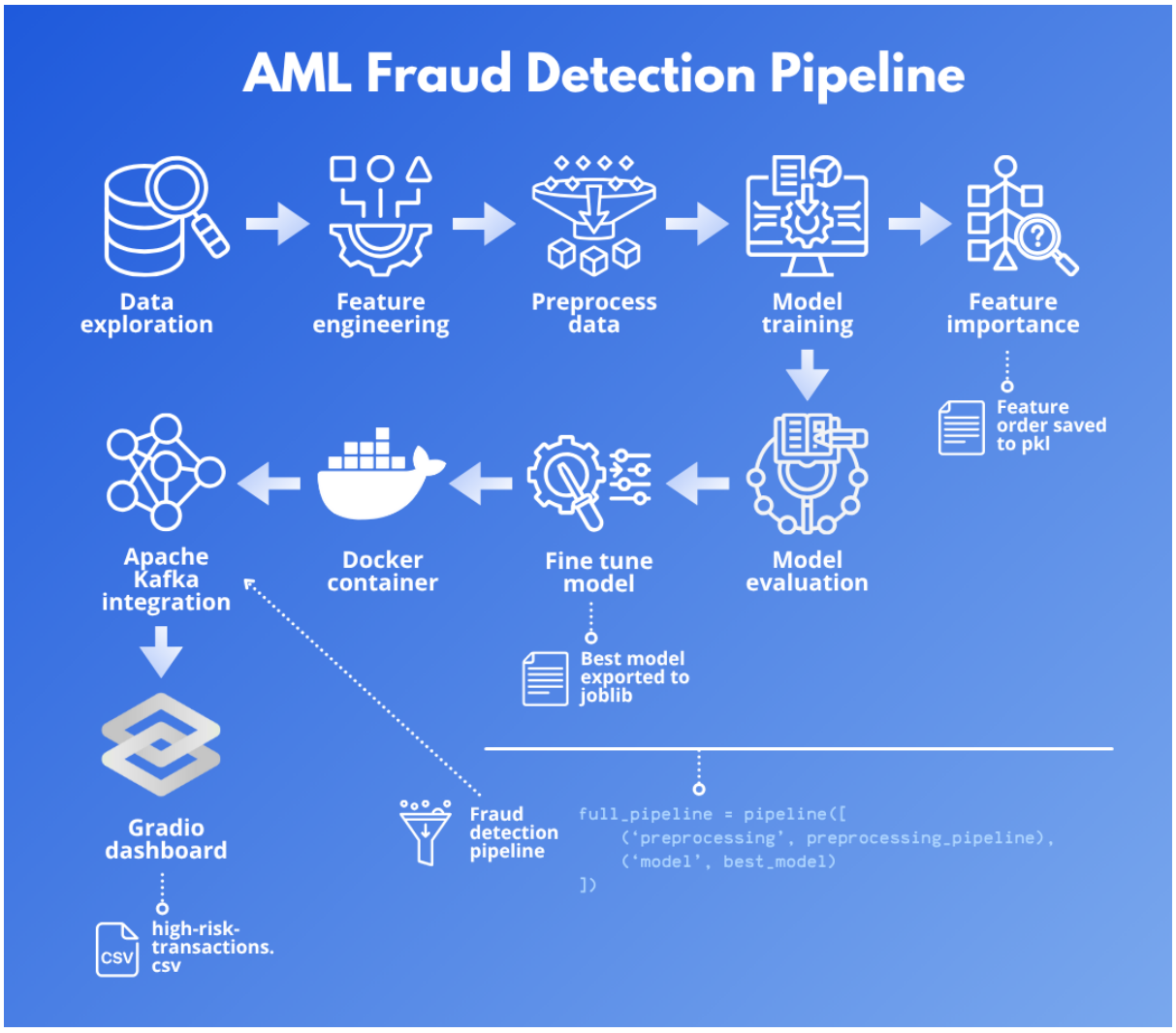

Figure 16: AML fraud detection pipeline

12. References

AML Watcher. (2024, June 20). *Top 9 Features of Efficient AML Software in 2024.* https://amlwatcher.com/blog/top-9-features-of-efficient-aml-software-in-2024/

Euvic. (n.d.). *FinTech compliance guide: How to navigate challenges and stay ahead.* https://www.euvic.com/us/post/fintech-compliance-guide/

LexisNexis. (2021, June 9). *Global spend on financial crime compliance at financial institutions.* https://risk.lexisnexis.com/global/en/about-us/press-room/press-release/20210609-tcoc-global-study

Trust Decision. (2024, June 6). *AI fraud detection: Identifying suspicious transactions.* https://trustdecision.com/resources/blog/ai-powered-fraud-detection-identifying-suspicious-transaction

United Nations Office on Drugs and Crime. (2024, November). *Improving regional investigations on money laundering and asset recovery.* https://www.unodc.org/roca/en/NEWS/news_2024/november/improving-regional-investigations-on-money-laundering-and-asset-recovery.html

Dataset and Code Resources

PaySim Dataset – Synthetic Financial Dataset for Fraud Detection

https://www.kaggle.com/datasets/ealaxi/paysim1/data

A synthetic dataset generated using the PaySim simulator that models mobile money transactions with injected fraudulent behaviour.

Code Repository

Project GitHub Repository

https://github.com/jenniet77/aml_kafka_proj/tree/main

Contains all code, models, and documentation for the AML Kafka project.

Exported Models

- `best_RandomForestClassifier.joblib` – final trained Random Forest model
- `feature_order.pkl` – serialized list of feature column names for consistent input
- `fraud_detection_model.pkl` – complete ML pipeline saved for streamlined integration

---

Notebooks

- `data_exploration.ipynb` – Initial EDA and dataset understanding
- `feature_engineering.ipynb` – Feature creation and transformation
- `model_training.ipynb` – Model selection, training and evaluation
- `fraud_detection_pipeline.ipynb` – Complete pipeline including feature engineering, feature scaling, and fitting saved model, dumped to `fraud_detection_model.pkl` via joblib

Technologies and Libraries

Programming & ML

- Python – Primary programming language
- Scikit-learn – Machine-learning library
- XGBoost – Gradient boosting library
- Numpy – Numerical computations
- Joblib – Model serialization

Data handling & EDA

- Pandas – Data manipulation library
- Matplotlib & Seaborn – Visualization libraries
- Jupyter Notebook – Used for exploratory data analysis and model development

Streaming & Infrastructure

- Docker & Docker Compose – Containerization
- Apache Kafka – Distributed streaming platform
- Kafka-Python – Kafka client in producer/consumer scripts
- Loguru – For structured logs in Kafka producer/consumer scripts

Interface & Presentation

- Gradio – Lightweight interactive web interface for demo/monitoring