**Mule Audit – Machine Learning–Based Detection Framework**

**1. Objective**

This project aims to identify potential mule accounts in the bank by analysing transactional behaviour and similarities with already confirmed mule accounts.  
The outcome will help the audit and analytics teams focus on high-risk behavioural clusters for detailed review and control testing.

**2. Data Preparation**

**2.1 Objective**

The purpose of this stage is to extract, cleanse, and organize all relevant data required for the mule account analytics model.  
The output should represent a consolidated, high-quality dataset that accurately reflects transactional and behavioural characteristics of each account — ready for clustering and subsequent audit analysis.

**2.2 Data Sources**

Relevant data typically spans multiple domains within the bank’s data warehouse. The key logical sources include: **(Field name will not match with the actual data, it is only for illustration purposes)**

| **#** | **Data Domain** | **Illustrative Table** | **Key Fields** | **Purpose** |
| --- | --- | --- | --- | --- |
| 1 | **Customer Master** | CUSTOMER\_MASTER | Customer\_ID, Customer\_Type, KYC\_Status, Nationality, Onboarding\_Channel, Risk\_Rating, Date\_Opened, Branch\_Code, Geography | Provides static profile data for each customer; supports segmentation and peer benchmarking. |
| 2 | **Account Master** | ACCOUNT\_MASTER | Account\_ID, Customer\_ID, Account\_Type, Account\_Status, Open\_Date, Close\_Date, Balance, Product\_Code | Enables linkage between customers and operational accounts; filters active accounts and provides balance-level insights. |
| 3 | **Transaction Ledger** | TRANSACTION\_DETAIL | Txn\_ID, From\_Account, To\_Account, Txn\_Amount, Txn\_Date, Txn\_Time, Txn\_Channel, Txn\_Type, Counterparty\_Bank, Narration | Core dataset capturing monetary movement patterns. Used for deriving behavioural, temporal, and velocity features. |
| 4 | **Known Mule Reference** | MULE\_REFERENCE\_LIST | Account\_ID, Source, Confirmation\_Date, Investigation\_ID | Ground truth dataset for evaluating clustering outcomes and identifying high-risk clusters. |
| 5 | **Device & Channel Data** | DEVICE\_ACCESS\_LOGS | Account\_ID, Device\_ID, IP\_Address, Login\_Time, Channel\_Type | Identifies accounts accessed from common devices or IPs; helps detect collusive or shared-infrastructure behaviour. |
| 6 | **Counterparty Summary** | COUNTERPARTY\_MASTER | Counterparty\_Account, Bank\_IFSC, Entity\_Type, Transaction\_Count, Total\_Value, High\_Risk\_Flag | Consolidates frequently interacting external accounts and helps trace mule-to-mule or mule-to-third-party linkages. |
| 7 | **Beneficiary Master** | BENEFICIARY\_MASTER | Beneficiary\_ID, Account\_ID, Beneficiary\_Account, Beneficiary\_IFSC, Add\_Date, Added\_By\_Channel, Approval\_Time, Txn\_Initiated\_Flag | Captures registered beneficiaries linked to each account; helps identify fake or rapidly changing beneficiary lists, shared beneficiaries across customers, and mule ring linkages. |
| 8 | **Branch & Staff Mapping** | BRANCH\_STAFF\_MAP | Branch\_Code, RM\_ID, Staff\_ID, Role, Access\_Privilege | Enables branch-level comparison and detection of unusual clusters linked to specific relationship managers or staff. |
| 9 | **Payment Channel Metadata** | CHANNEL\_MASTER | Channel\_Code, Description, Transaction\_Limit, Processing\_Time, Risk\_Score | Supports channel-wise normalization and helps assess risk associated with specific payment types. |

**2.3 Data Extraction**

**2.3.1 Extraction Strategy**

Data extraction follows a phased approach to ensure performance, consistency, and quality validation before scaling to the enterprise level.

**Phase 1 – Exploration (Pilot Month)**

* Extract approximately **one month of transactional data** across all active accounts.
* This enables validation of joins, transformations, and feature logic before full-scale processing.
* Conduct key checks on record counts, timestamp integrity, field completeness, and linkage success rates.

**Phase 2 – Scale-Up (Full Historical Window)**

* Upon validation, extend extraction to **12–24 months** of historical data.
* Maintain temporal continuity to capture seasonal trends, dormant-to-active cycles, and periodic inflow bursts.
* Include closed or inactive accounts, as closure timing can itself indicate potential mule behaviour.

**2.3.2 Source Integration and Joining Logic**

The extraction layer consolidates heterogeneous datasets into a unified analytical base. **(This is only as an example, actual table mapping is available in the data dictionary)**

| **Join Key** | **Data Sources Involved** | **Join Type** | **Purpose** |
| --- | --- | --- | --- |
| Account\_ID | Transaction ↔ Account ↔ Mule List ↔ Device Logs ↔ Alert History | Inner / Left | Binds all activity and attributes to account-level entities. |
| Customer\_ID | Customer Master ↔ Account Master | One-to-Many | Enables customer-level aggregation across multiple accounts. |
| Branch\_Code | Account Master ↔ Branch Staff Map | Left | Adds branch and staff association context. |
| Device\_ID / IP\_Address | Device Logs ↔ Transaction Table | Inner | Links accounts accessed from the same digital footprint. |

Integration must ensure:

* One unique record per transaction post-join.
* Foreign key integrity (no orphaned transactions or customers).
* Consistent field names, data types, and date formats across all sources.
* Auditability through a **join mapping log** documenting record-level merge success or exclusion.

**2.3.3 Data Volume and Performance Considerations**

* Apply **date partitioning** to extract large transaction tables incrementally.
* Use **batch identifiers** or **extraction timestamps** for refresh control.
* Maintain versioned extracts (v1.0, v1.1, etc.) for traceability.
* Store intermediate datasets (e.g., monthly summaries) in a **data mart schema** for reuse and reproducibility.

**3. Feature Engineering**

Feature engineering occurs in two structured stages to maintain separation between transaction-level computations and cross-domain aggregation.

**Stage 1 — Transaction-Level Feature Derivation**

Behavioural features are first derived from the transaction ledger before merging with any other datasets.  
Representative examples include: **(This is not an exhaustive list, you can add many more to it if needed. I have tried to explain why these features are needed, which might be helpful while creating new features)**

| **#** | **Feature Category** | **Example Feature** | **Definition / Description** | **Audit Relevance** |
| --- | --- | --- | --- | --- |
| 1 | **Transaction Volume** | total\_inflow, total\_outflow, txn\_count | Sum and count of all credits and debits per account. | High throughput accounts may indicate layering or conduit behaviour. |
| 2 | **Velocity** | avg\_txn\_per\_day, max\_txn\_in\_a\_day, inter\_txn\_gap\_mean | Frequency and intensity of transactions within defined time windows. | Identifies burst activity or high-turnover accounts. |
| 3 | **Directionality Ratio** | inflow\_outflow\_ratio, credit\_to\_debit\_ratio | Measures balance between received and transferred funds. | Ratios near 1 suggest quick pass-through patterns. |
| 4 | **Counterparty Spread** | unique\_counterparties, repeat\_counterparty\_ratio | Count of unique and recurring counterparties. | Mule rings often reuse a fixed set of linked accounts. |
| 5 | **Temporal Behaviour** | night\_txn\_ratio, weekend\_txn\_ratio, month\_end\_activity\_ratio | Distribution of transactions across time-of-day and calendar periods. | Off-hour and weekend spikes suggest concealed or automated transfers. |
| 6 | **Circular Flow Indicators** | self\_loop\_count, two\_hop\_return\_rate, fund\_reversal\_ratio | Tracks money looping back to the originator within 1–2 hops. | Key red flag for mule layering cycles. |
| 7 | **Transaction Size Variation** | avg\_txn\_amount, std\_txn\_amount, small\_value\_txn\_ratio | Mean, deviation, and proportion of micro-value transactions. | Micro-transfers are often used to test mule pipelines. |
| 8 | **Dormancy and Reactivation** | days\_inactive\_before\_burst, txn\_reactivation\_flag | Time gap between periods of inactivity and sudden bursts. | Detects dormant accounts repurposed for mule activity. |
| 9 | **Balance Behaviour** | avg\_balance\_post\_txn, balance\_decay\_rate | Post-transaction balance trends. | Mule accounts tend to zero-out balances after each flow. |
| 10 | **Peer Deviation Score** | branch\_peer\_zscore, segment\_peer\_zscore | Z-score comparing an account’s metrics to its branch or segment peers. | Quantifies behavioural outliers within similar cohorts. |
| 11 | **Geo or Channel Concentration (if available)** | channel\_diversity\_index, geo\_txn\_concentration | Ratio of distinct transaction channels or geographies used. | Multi-channel transfers or frequent cross-geo jumps are risk indicators. |
| 12 | **High-Risk Transaction Ratio** | hr\_txn\_ratio | % of transactions involving flagged or high-risk counterparties. | Highlights exposure to known risky entities or networks. |

These metrics are stored in an intermediate table, e.g., TXN\_FEATURE\_SUMMARY.

**Stage 2 — Cross-Domain Aggregation and Enrichment**

After transaction-level metrics are finalized:

1. **Aggregate to Account Level**
   * Summarize behavioural indicators by Account\_ID.
   * Join with Account Master and Customer Master to append product, KYC, geography, and segment information.
2. **Integrate Additional Dimensions**
   * **Device Linkage:** number of unique and shared devices per account.
   * **Branch/Staff Association:** count of high-risk accounts per RM or staff member.
   * **Alert History:** count or frequency of prior AML/fraud alerts.
   * **Mule Reference:** add the binary flag is\_known\_mule for known accounts.
3. **Analytical Base Table (ABT)**
   * The consolidated dataset, typically ACCOUNT\_FEATURE\_MASTER, contains one record per account.
   * Maintain feature prefixes (FTXN\_, FDEV\_, FBR\_, etc.) for traceability.
   * Document all transformations and derivations for audit transparency.

**Outcome of Data Preparation and Feature Engineering**

* Validated, join-consistent dataset spanning multiple domains.
* Derived behavioural metrics enriched with customer, device, and alert dimensions.
* Ready input for exploratory data analysis and unsupervised clustering.

**4. Modelling Approach**

**4.1 Objective**

The modelling stage segments accounts into behaviourally homogeneous clusters using unsupervised learning techniques.  
These clusters reveal transaction patterns, velocity profiles, and linkage characteristics that highlight accounts exhibiting potential mule behaviour.

**4.2 Model Selection Rationale**

Confirmed mule cases represent a small fraction of total accounts, creating severe class imbalance.  
Therefore, **unsupervised clustering** allows the data to naturally group based on behavioural similarity without pre-labelled outcomes.

| **Algorithm** | **Type** | **Key Characteristics** | **Typical Use Case** |
| --- | --- | --- | --- |
| **K-Means** | Centroid-based | Fast, scalable, assumes spherical clusters | Baseline segmentation; high interpretability. |
| **DBSCAN** | Density-based | Detects arbitrary-shaped clusters and isolates noise | Effective for outlier or high-velocity accounts. |
| **HDBSCAN** | Hierarchical density-based | Handles variable densities and nested clusters | Useful for diverse customer segments and transaction scales. |

A **tiered approach** is recommended:

1. Begin with **K-Means** for initial segmentation.
2. Re-run using **DBSCAN** or **HDBSCAN** to uncover additional anomaly clusters.
3. Compare results for stability and interpretability.

**4.3 Pre-Modelling Preparation**

1. **Variable Selection**
   * Use numeric, ratio, and continuous features from ACCOUNT\_FEATURE\_MASTER.
   * Remove redundant or highly correlated variables (|r| > 0.8).
   * Exclude identifiers or categorical codes unless encoded.
2. **Standardization and Scaling**
   * Standardize features to zero mean and unit variance using z-score normalization.
   * Apply log transformation to mitigate skewness in heavy-tailed variables such as transaction values.
3. **Dimensionality Reduction (Optional) 🡪 If you end more than 40 features then use this step or else ignore it. Using this might lead to difficulty while explaining the model results.** 
   * Use **PCA** for visualization or to remove multicollinearity.
   * Retain components explaining 85–90% of variance when feature count exceeds 30.
4. **Data Integrity Checks**
   * Validate that all records contain complete standardized values.
   * Confirm consistency of metrics before and after scaling.

**4.4 Model Execution**

1. **K-Means Baseline**
   * Identify optimal *k* using elbow, silhouette, and Calinski–Harabasz criteria.
   * Fit the model and assign cluster labels to each account.
   * Examine centroids to interpret behavioural themes (e.g., low-activity vs. burst-transfer clusters).
2. **DBSCAN / HDBSCAN Validation**
   * Run DBSCAN to detect dense pockets and outlier accounts.
   * Tune eps (radius) and min\_samples to balance cluster sensitivity.
   * Use HDBSCAN for mixed-density scenarios.
   * Compare overlaps between density-based and centroid-based clusters for robustness.
3. **Stability Testing**
   * Re-run models on random 80% samples.
   * Compute **Adjusted Rand Index (ARI)** or **Normalized Mutual Information (NMI)** to assess cluster stability.

**4.5 Mule Overlay and Cluster Evaluation**

1. **Overlay Known Mules**
   * Map the binary field is\_known\_mule to cluster assignments.
   * Compute mule density for each cluster:

Mule Density (%)=Known Mules in ClusterTotal Accounts in Cluster×100\text{Mule Density (\%)} = \frac{\text{Known Mules in Cluster}}{\text{Total Accounts in Cluster}} \times 100Mule Density (%)=Total Accounts in ClusterKnown Mules in Cluster​×100

1. **Rank Clusters by Risk (percentages below are just example it has to be discussed with Gowtham/Richard before considering it)**
   * Sort clusters by mule density:
     + **High Risk:** > 10% known mules
     + **Medium Risk:** 10–20%
     + **Low Risk:** < 5%
   * Visualize results via bar charts or heatmaps.
2. **Behavioural Interpretation**
   * Review key features for high-risk clusters.
   * Common red flags include:
     + High inflow/outflow ratios with minimal retention
     + Rapid sequential transactions across linked accounts
     + Multiple accounts sharing devices or IPs
     + Concentrated branch or time-window activity
   * Record behavioural observations for audit sampling.

**4.6 Model Validation and Sensitivity Analysis**

* **Internal Validation:** Silhouette Score, Davies–Bouldin Index.
* **External Validation:** Mule overlay density and analyst review.
* **Sensitivity Testing:** Exclude specific feature groups (e.g., device metrics etc., ) to assess risk rank shifts.
* **Explainability:** Maintain the top 5 to 8 contributing features per cluster for audit transparency.

**4.7 Deliverables**

1. **Cluster Assignment File:** Account-level list with cluster ID and mule density ranking.
2. **Model Summary:** Algorithms, parameters, and performance metrics.
3. **Cluster Profiles:** Summary statistics and behavioural themes per cluster.

| **Phase** | **Description** | **End Date** |
| --- | --- | --- |
| **1. Data Extraction – Pilot Month** | Extract one month of transaction data for testing joins and feature logic. |  |
| **2. Data Cleansing & Validation** | Perform data quality checks, remove duplicates, and validate joins across sources. |  |
| **3. Feature Engineering – Transaction Level** | Create key behavioural and transactional features from raw transaction data. |  |
| **4. Feature Enrichment – Cross Domain** | Merge customer, beneficiary, and device dimensions to build a consolidated feature base. |  |
| **5. Exploratory Data Analysis (EDA)** | Analyse distributions, correlations, and outlier patterns to understand account behaviour. |  |
| **6. Clustering Model Development** | Apply K-Means for baseline clustering and validate results using DBSCAN/HDBSCAN. |  |
| **7. Mule Overlay & Risk Evaluation** | Overlay known mule list, compute mule density, and identify high-risk clusters. |  |
| **8. Cluster Review & Interpretation** | Analyse high-risk clusters for behavioural patterns and document key observations. |  |