**FX Anomaly Model — Retraining & Onboarding (Databricks + Azure Blob)**

This page explains **two tracks** for our autoencoder-based anomaly model on Databricks:

* **Track A — Existing clients:** scheduled retraining every 60 days.
* **Track B — New client onboarding:** gated process to align new data to our pipeline.

We keep this practical and developer‑focused while using correct ML terms.

**1) Overview & Assumptions**

* **Model:** Reconstruction‑error **autoencoder** trained on ~4 years of historical FX transactions.
* **Goal:** Improve anomaly detection while keeping training deterministic and easy to reproduce.
* **Platform:** Databricks Jobs/Workflows + MLflow; data in **Azure Blob Storage**.
* **Evaluation:** Use a **fixed holdout test set** for acceptance (no leakage). Primary metric = **F1**.

**2) Data & Schedule**

**Blob layout per client**

<client\_name>/

History/ # initial bulk loads & backfills for onboarding (Track B)

<ingest\_timestamp>/

\*.parquet

retraining/ # periodic drops for retraining (Track A)

<ingest\_timestamp>/

\*.parquet

test\_set/ # fixed, labeled holdout for acceptance testing only

<version\_or\_asof>/

\*.parquet

* <ingest\_timestamp>: yyyyMMdd\_HHmmss (e.g., 20250115\_000000).
* History/ and retraining/ are **append‑only**; prior timestamps are immutable.
* test\_set/ is **append‑only and versioned**; never mixed with train/val data; used only to compute baseline and candidate F1.

**Cadence & triggers (Track A)**

* Target **every 60 days** per client.
* Two ways to start a run:
  1. **Event‑driven:** upstream sends a message/email that new data is available under retraining/.
  2. **Time‑driven poller:** after day 60, check for new data up to **4 attempts** at **6‑hour** intervals. If nothing new is found, mark **Missed Window** and alert.

**3) Common Rules (both tracks)**

* **Schema contract** (minimum columns; examples):
  + IDs/time: txn\_id, txn\_ts, client
  + Categorical: Instrument, BUnit, Cpty, BuySellTypeID, PrimaryCurr, BuyCurr, SellCurr
  + Numeric: BuyAmount, SellAmount, BuyBalanceMovement, SellBalanceMovement, derived ratios
* **Data hygiene:** standardize missing markers (e.g., "..", "N/A") to **null**; parse timestamps to UTC.
* **Duplicates:** no duplicate txn\_id within a drop after de‑duplication (latest by txn\_ts wins).
* **Validation (fail fast):**
  + Required columns missing or wrong dtype
  + Row count below min\_rows\_per\_window (client‑configurable; default 50k)
  + 5% duplicates post de‑duplication
* **Validation (warn and proceed):** higher‑than‑expected nulls, extreme outliers, unusual category explosion.

**Exit Points & Outcomes (both tracks)**

* **FAILED\_SANITY**: Schema/type checks fail or unparseable timestamps → **stop before training**; notify data provider with failed checks.
* **FAILED\_VOLUME**: After sanity passes, effective rows < min\_rows\_per\_window (post de‑dup/filtering) → **stop before training**; notify provider.
* **REJECTED\_NO\_IMPROVEMENT**: Training completes but ΔF1 < 0.01 on the fixed test\_set/ → **no deployment**; production model remains unchanged.
* **SKIPPED\_NO\_DATA**: No new data found after polling window → **no run** beyond logging/alerting.

**4) Track A — Existing Clients: Retraining Flow**

**Intake**

* Read all new files under <client\_name>/retraining/<ingest\_timestamp>/.
* Merge, de‑duplicate, validate; log run inputs (file list, sizes, hashes).

**Training window & features**

* Use a **rolling 4‑year window** ending at the newest txn\_ts in the drop.
* Feature pipeline (re‑fit each run and versioned with the model):
  + Numeric scaling (Standard/Robust)
  + Categorical encoding (OHE or embeddings with <UNK> bucket)
  + Date‑derived features (durations, calendar features) as configured
  + Optional clipping/log for heavy‑tailed amounts

**Model training**

* Autoencoder with fixed architecture/hyperparameters (same as the deployed configuration).
* Early stopping on validation loss; threshold chosen on validation to optimize **F1**.

**Evaluation & acceptance**

* Evaluate on the **fixed holdout test set** in test\_set/.
* **Promote** only if F1\_new − F1\_baseline ≥ 0.01 (≥ **1 percentage point** absolute improvement).
* If not met, mark run **REJECTED\_NO\_IMPROVEMENT**; **no deployment** and production stays on the current version.

**Deployment**

* Register artifacts: model, preprocessors, threshold.json, schema.json, training\_manifest.json.
* Transition via MLflow **Staging → Production**. Use a brief canary if required by the client.

**Failure cases**

* **SKIPPED\_NO\_DATA**: No new data after 4 polls → log and alert; no training.
* **FAILED\_SANITY**: Sanity/schema checks fail → stop before training; share failed expectations.
* **FAILED\_VOLUME**: Row count below threshold after sanity/de‑dup → stop before training.
* **FAILED\_TRAINING**: Exception during training → log; auto‑retry once on a clean cluster; alert if persistent.
* **REJECTED\_NO\_IMPROVEMENT**: ΔF1 below 0.01 on test\_set/ → no deployment; production unchanged.

**5) Track B — New Client Onboarding**

**Data drop**

* Provider writes bulk history to <client\_name>/historical/<ingest\_timestamp>/\*.parquet.
* Minimum volume: configurable (e.g., ≥12 months or ≥250k rows).

**Automated checks**

* Run the same validation suite as Track A (schema, dtypes, nulls, duplicates, ranges).
* Compute a **compatibility score** against our reference feature set (presence of mandatory fields, basic unit conventions).

**EDA & mapping (human‑in‑the‑loop)**

* Quick profiling: distributions, missingness, categorical coverage, and domain sanity checks (e.g., BuyAmount vs SellAmount per currency).
* Map client columns to our **featureset v1** (or define featureset v2 if needed). Document transformations.

**First‑cut model**

* Where compatible, train using the **same pipeline and hyperparameters** as existing clients.
* Evaluate on a labeled sample (if available) or rules‑based backtest to estimate precision/recall.
* This is a starting point; further tuning may be required before Production.

**Decision**

* If compatibility and metrics meet minimums, proceed to deployment (with canary if desired).
* Otherwise, provide an onboarding gap report and recommended data or feature changes.

**6) Operations & Governance**

* **Config:** Central clients.yaml stores per‑client thresholds (e.g., min\_rows\_per\_window, cadence days, acceptance delta F1).
* **Secrets:** Use Databricks Secret Scopes; never hardcode credentials.
* **Auditability:** Every run logs parameters, metrics, code commit SHA, and a signed training manifest.
* **Versioning:**
  + Retrain with unchanged architecture/HPs → **MINOR** bump
  + Preprocessor/bugfix only → **PATCH**
  + Architecture/HP change → **MAJOR** (outside routine retraining)

**7) Quick Runbook**

* **Force a retrain**: run the Track‑A job with client\_name + a specific ingest\_timestamp.
* **Rollback**: switch the previous MLflow registry version to **Production**.
* **Add a client**: create blob paths, update clients.yaml, set secrets, run Track‑B checks, then first‑cut model.

**8) Definitions**

* **Reconstruction error**: difference between input and autoencoder output; higher error suggests anomaly.
* **Holdout**: fixed test set never used in training/validation; used only for acceptance.
* **Acceptance delta (ΔF1)**: minimum absolute improvement (≥ 1 percentage point) required to promote a new model.