Governance-Grade Evidence for KPI Risk Under AI-Optimized Call Center Dynamics:
NovaFabric Validation Checklist (Synthetic Case Study)

Gina Aulabaugh

February 17, 2026

Abstract

This paper documents a governance-grade validation workflow for a synthetic call-center dataset ("NovaFabric") designed to support audit-friendly claims about KPI behavior under AI-optimized operational pressures. The core contribution is an evidence chain: deterministic instrumentation, file hashing, run-stamped outputs, and repeatable validation artifacts (decile lift tables, time-series stability views, and logistic regression odds ratios). Results show monotonic relationships between friction and three operational outcomes (ticketing, repeat contact within seven days, and resolution), producing paper-proof artifacts suitable for replication and review.

1 Introduction

When operational metrics become targets, they stop behaving like measurements and start behaving like incentives. In call-center environments influenced by automation, routing optimization, and emerging AI assistance, governance requires more than dashboard summaries. It requires proof: what was run, on what data, what transformations occurred, and which outputs were produced.

This paper uses **NovaFabric** as an experimental platform and the **NovaFabric Validation Checklist** as a reproducible validation pipeline. The objective is to generate **audit-grade evidence** about relationships among friction signals and downstream outcomes, while keeping the work suitable for external review. The emphasis is not on claiming real-world causality, but on producing a chain of artifacts that a reviewer can trace end-to-end.

2 Artifact Index

All artifacts referenced in this paper are produced under:

`output/novafabric_validation/`

Table 1 lists the exact filenames referenced by this paper.

Table 1: Artifact Index (exact filenames)

Filename	Purpose / Paper Role
`NovaFabric_instrumented.csv`	Instrumented dataset used for all validation computations (filled columns, standardized bucket labels).
`instrumentation_receipt.json`	Receipt describing instrumentation decisions (what columns were created/used and why).
`input_sha256.txt`	SHA-256 hash of the validated input (tamper-evidence for the exact bytes analyzed).
`validation_summary.json`	Machine-readable summary of checks, thresholds, warnings, and selected columns.
`validation_report.md`	Human-readable report for reviewers (tables, warnings, existence checks).
`validation_metrics_by_bucket.csv`	Outcome rates by standardized subreason bucket (volume and means).
`validation_metrics_by_rep.csv`	Outcome rates by representative (rep-level behavior and outlier detection).
`decile_lift_table.csv`	Primary results table: ticket/resolved/repeat rates by friction decile.
`model_summary.json`	Summary of model settings and sample sizes for logistic regressions.
`logit_ticket_uncontrolled_oddsratios.csv`	Odds ratios for `ticket_flag` using friction alone (baseline model).
`logit_ticket_controls_oddsratios.csv`	Odds ratios for `ticket_flag` controlling for `subreason` and `rep_id`.
`logit_resolved_uncontrolled_oddsratios.csv`	Odds ratios for `resolved_flag` using friction alone.
`logit_resolved_controls_oddsratios.csv`	Odds ratios for `resolved_flag` controlling for `subreason` and `rep_id`.
`plots/run_<runid>/*.png`	Run-isolated figures to prevent stale plots being mistaken for new results.

3 Methods

This section is intentionally written in plain English first, because governance workflows fail most often at the handoff between technical truth and human understanding.

3.1 Project Structure and Reproducibility

The project is repo-root compliant: scripts locate the repository root (presence of `data/` and `src/`) and write outputs to a fixed destination (`output/novafabric_validation/`). This design allows the workflow to run from any working directory without silently writing outputs to unexpected locations.

Two scripts form the pipeline:

- `src/00_instrument_novafabric.py` instruments the dataset into stable "filled" columns and emits an instrumentation receipt.
- `src/01_validate_novafabric.py` performs validation checks, generates plots and tables, hashes inputs, and emits run-stamped outputs.

3.2 Integrity Chain

The integrity chain is designed so a reviewer can answer four uncomfortable questions: (1) What data did you validate? (2) What transformations did you apply? (3) What exactly did the run produce? (4) Could outputs be stale or swapped?

Step 1: Instrumentation (make the data stable). Real datasets often contain missingness, inconsistent naming, and partial fields. Instrumentation standardizes raw inputs into predictable fields such as `friction_level_filled`, `trust_score_filled`, `ticket_flag_filled`, and `resolved_flag_filled`. This reduces ambiguity in downstream computations and prevents accidental column drift.

Step 2: Receipt (prove what transformations were applied). Instrumentation writes `instrumentation_rece` describing which columns were created, selected, or standardized. This receipt functions as a run record for audit and reproducibility.

Step 3: Hashing (prove what bytes were analyzed). Validation computes a SHA-256 hash of the input and stores it in `input_sha256.txt`. If the input file changes later, the hash changes, providing tamper-evidence for the validated bytes.

Step 4: Run isolation (prove the run produced these outputs). Plots are written into run-specific directories (`plots/run_<runid>/`) rather than overwriting old figures. This eliminates stale-image confusion during iterative runs.

3.3 Validation Checks

The validator performs coverage and missingness gates, range and sanity checks, time-series stability scans, and behavioral validity tests using friction-decile lift (`decile_lift_table.csv`) and logistic regression odds ratios (CSV outputs summarized by `model_summary.json`).

4 Results

4.1 Distribution and Stability Evidence

Friction and trust distributions indicate continuous, plausible variation suitable for downstream analysis. Daily time series provide a stability scan: call volume stays within a consistent band while friction and outcome rates vary within interpretable ranges.

4.2 Primary Finding: Friction Aligns With Operational Outcomes

Friction-decile lift behavior computed from `decile_lift_table.csv` shows that higher friction corresponds to higher ticket rate, higher repeat contact within 7 days, and lower resolution rate. This supports friction as an upstream risk signal within the synthetic environment.

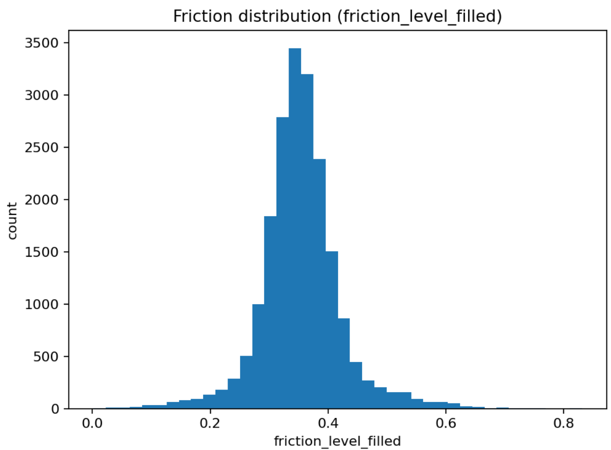
4.3 Model-Based Confirmation

Logistic regression odds ratio outputs quantify the friction effect:

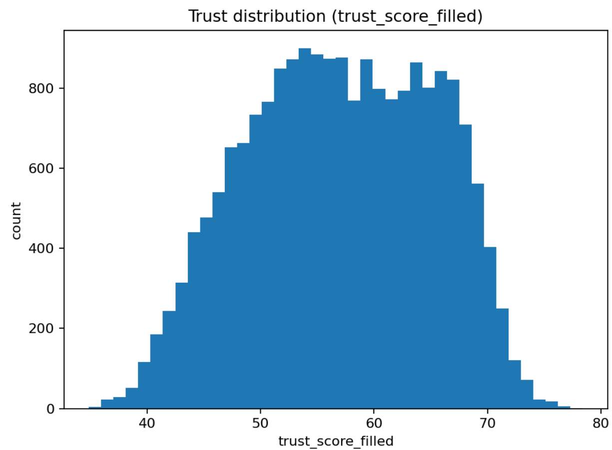
- Baseline: `logit_ticket_uncontrolled_oddsratios.csv`, `logit_resolved_uncontrolled_oddsratios.csv`
- Controlled: `logit_ticket_controls_oddsratios.csv`, `logit_resolved_controls_oddsratios.csv`

Controlled models reduce the risk that observed relationships are solely due to subreason mix or representative-specific behavior. Model configuration and sample sizes are summarized in `model_summary.json`.

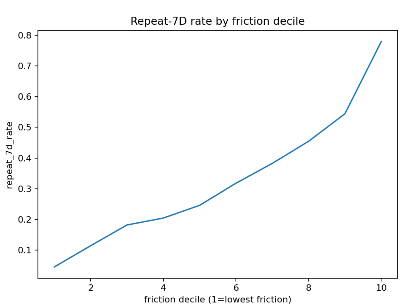
5 Results Figures



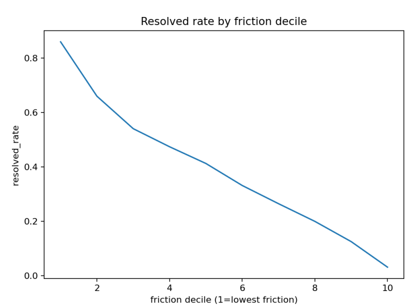
(a) Friction distribution



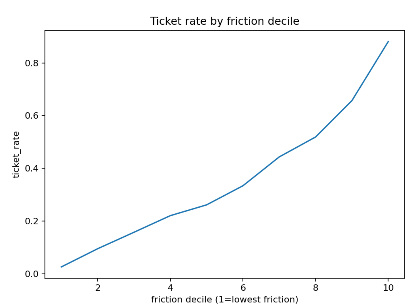
(b) Trust distribution



(c) Repeat-7D by friction decile

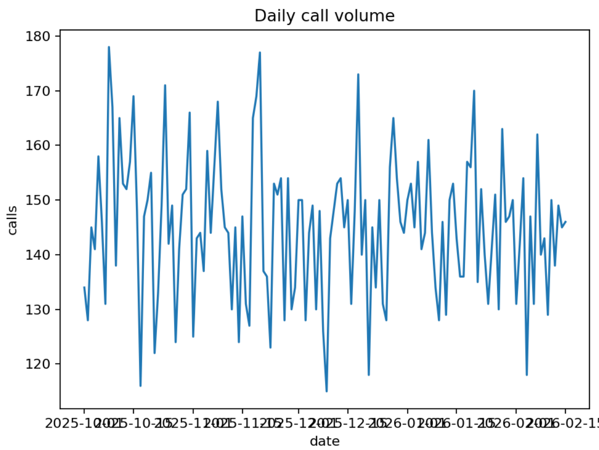


(d) Resolved by friction decile

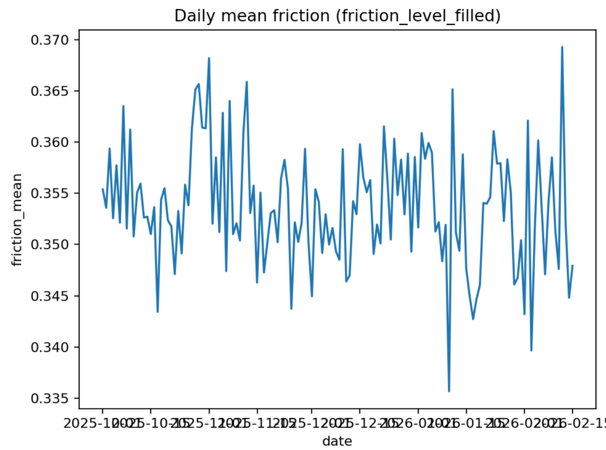


(e) Ticket by friction decile

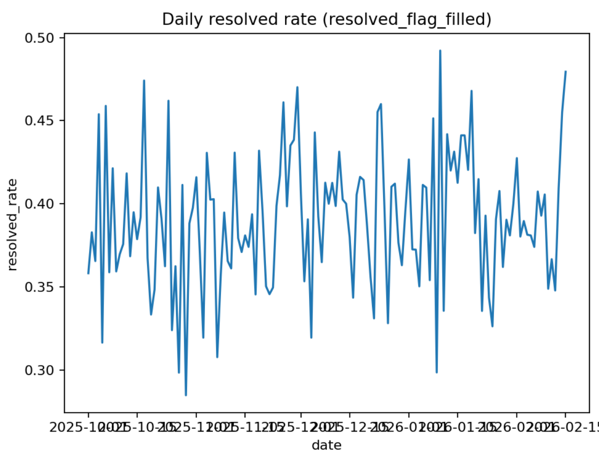
Figure 1: Distributions and monotonic lift evidence. Lift curves are computed from `decile_lift_table.csv`.



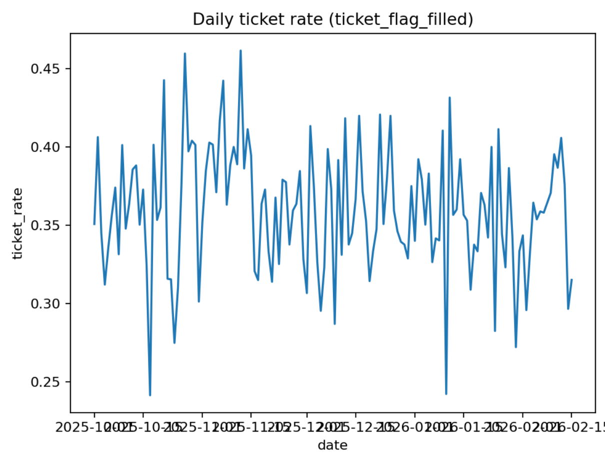
(a) Daily call volume



(b) Daily mean friction



(c) Daily resolved rate



(d) Daily ticket rate

Figure 2: Time-series stability scan for volume, friction, and key outcome rates.

6 Discussion and Limitations

This workflow treats descriptive plots as diagnostics, not proof. The proof is in monotonic lift behavior, interpretable effect sizes, and a receipt chain that makes each run reproducible. Because this study uses synthetic data, results should not be interpreted as empirical claims about any specific real-world call center. The contribution is the governance method: receipt-based auditability, filled-column integrity gates, and interpretable evidence artifacts suitable for replication.

7 Conclusion

This paper documents a governance-grade evidence chain for validating KPI-related risk signals in a synthetic call-center environment. The integrity chain (`instrumentation_receipt.json`, `input_sha256.txt`, run-isolated plots, and structured validation reports) reduces the risk of stale outputs and increases auditability. Results support friction as an upstream risk signal: higher friction aligns with higher ticketing and repeat contact and lower resolution, producing paper-proof artifacts for governance critique work under AI-accelerated optimization.

References

[1] FCC. (n.d.). *CGB – Consumer Complaints Data* [Data set]. Retrieved February 16, 2026, from https://opendata.fcc.gov/Consumer/CGB-Consumer-Complaints-Data/

[2] U.S. General Services Administration. (n.d.). *Call Center Metrics for the Health Service System* [Data set]. Data.gov. Retrieved March 15, 2026, from https://catalog.data.gov/dataset/call-center-metrics-for-the-health-service-system

[3] U.S. General Services Administration. (n.d.). *MTA Customer Engagement Statistics (Beginning May 2017)* [Data set]. Data.gov. Retrieved February 15, 2026, from https://catalog.data.gov/dataset/mta-customer-engagement-statistics-beginning-may-2017

[4] U.S. General Services Administration. (n.d.). *MTA Access-A-Ride Call Center Performance (Beginning 2016)* [Data set]. Data.gov. Retrieved February 15, 2026, from https://catalog.data.gov/dataset/mta-access-a-ride-call-center-performance-beginning-2016

[5] IBM. (n.d.). *Telco Customer Churn* [Data set]. Retrieved February 15, 2026, from https://raw.githubusercontent.com/IBM/telco-customer-churn-on-icp4d/master/data/Telco-Customer-Churn.csv

[6] Satvicoder. (n.d.). *Call Center Data* [Data set]. Kaggle. Retrieved March 15, 2026, from https://www.kaggle.com/datasets/satvicoder/call-center-data

[7] Ziya07. (n.d.). *Employee Churn Data* [Data set]. Kaggle. Retrieved March 15, 2026, from https://www.kaggle.com/datasets/ziya07/employee-churn-data

[8] Mexwell. (n.d.). *Employee Performance and Productivity Data* [Data set]. Kaggle. Retrieved March 15, 2026, from https://www.kaggle.com/datasets/mexwell/employee-performance-and-productivity-data

[9] Prishatank. (n.d.). *Employee HR Dataset* [Data set]. Kaggle. Retrieved March 15, 2026, from https://www.kaggle.com/datasets/prishatank/employee-hr-dataset