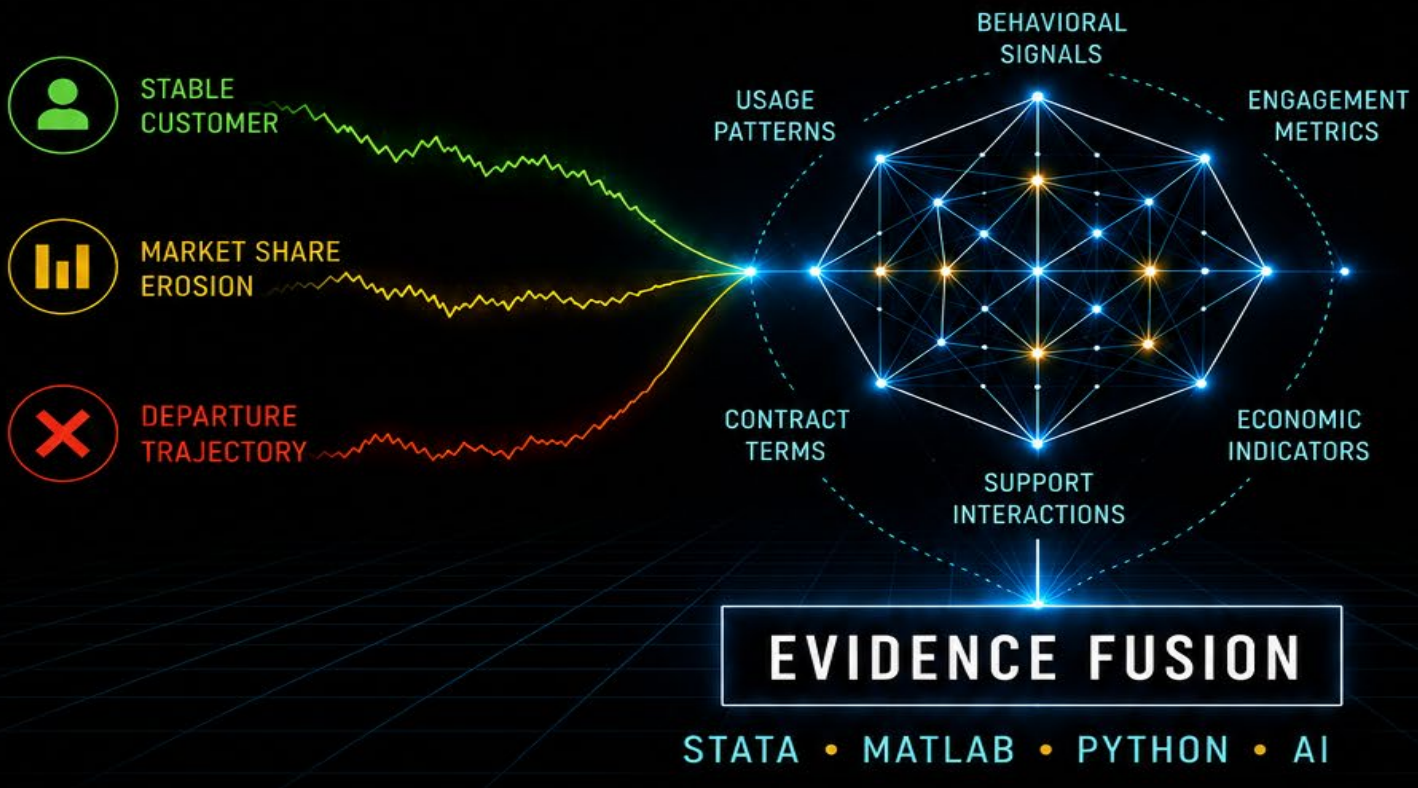
RATIO-WEEKLY RESEARCH

BEYOND DATA CLEANUP

Simulation, Trajectory Learning, and Evidence Fusion as a Governance-First Path to Enterprise AI



AN EMPIRICAL STUDY IN ENTERPRISE AI DEVELOPMENT

JOHN AARON, PHD

WWW.RATIO-WEEKLY.COM

2026

RATIO

AI TESTING & VALIDATION | PROJECT QUALITY & RISK MANAGEMENT

BEYOND DATA CLEANUP

This publication is intended to provide information and insights regarding artificial intelligence governance, simulation-driven development, customer behavior analytics, evidence fusion, probabilistic decision systems, and enterprise transformation. While every effort has been made to ensure the accuracy of the information presented, the author makes no representations or warranties regarding completeness, suitability, or fitness for a particular purpose. The author shall not be liable for any losses, damages, or decisions arising from the use of information contained in this publication. The views expressed herein are those of the author and do not necessarily reflect the views of any affiliated organizations, clients, or partners.

Title: Beyond Data Cleanup
Subtitle: Simulation, Trajectory Learning, and Evidence Fusion as a Governance-First Path to Enterprise AI
Author: John Aaron, PhD
Series: RATIO-WEEKLY RESEARCH
Website: www.ratio-weekly.com
Edition: First Edition, 2026
Distribution: Digital — ratio-weekly.com
Printed: United States of America

ARTIFICIAL INTELLIGENCE ASSISTANCE NOTICE

This publication was developed using a governed multi-platform research environment incorporating human expertise together with artificial intelligence systems. Artificial intelligence tools assisted with software development, simulation design, feature engineering, statistical analysis, visualization, documentation support, editorial refinement, and research acceleration.

Throughout the project, AI systems functioned as analytical assistants operating under human supervision and formal governance controls. Multiple AI platforms contributed at different stages of development, including support for Stata, MATLAB, Python, economic scoring frameworks, audit-file generation, visualization concepts, and manuscript preparation.

All research objectives, methodological decisions, interpretations, conclusions, economic assumptions, governance determinations, and final editorial decisions remained under human direction and responsibility. The author assumes full responsibility for the contents of this publication.

Beyond Data Cleanup

Simulation, Trajectory Learning, and Evidence Fusion as a Governance-First Path to Enterprise AI

An Empirical Multi-Platform Study in B2B Customer Departure Detection

Authors: John Aaron, PhD ·
Organization: Milestone Planning and Research, Inc. · RATIO AI Audit Practice
Location: Hickory Hills, Illinois
Date: June 2026
RATIO-WEEKLY Research Report · Version 17 · June 2026 · Proof of Concept
Contact: john.a@mprteam.com ·

Core Argument

Organizations spend millions of dollars attempting to clean, label, and integrate historical data before beginning AI development. This paper documents a governance-first alternative: a simulation-driven development path in which synthetic behavioral episodes, domain expertise, and rigorous economic scoring replace the dependency on large volumes of historically labeled data. The central finding is that an organization that can describe likely failure or departure patterns often already possesses the knowledge required to train a supervised AI system — without waiting for perfect historical records.

Abstract

Organizations commonly spend large sums attempting to clean, label, and integrate historical data before beginning artificial intelligence development. This paper documents a contrasting development path: a governance-centered, simulation-driven AI exercise in which synthetic weekly customer departure episodes were used to model likely future behavior rather than laboriously reconstruct a reliable historical record. The project began with prompt-only reasoning and conventional anomaly detection, moved through Stata, MATLAB, and Python implementations of behavioral surveillance models, and ultimately produced a hybrid evidence-fusion architecture that combines broad surveillance with precision trajectory recognition.

The dataset consisted of simulated weekly customer time series for 175 B2B customers observed over 156 weeks, producing 27,300 customer-week observations. Customers were segmented into continuous purchasers, periodic gap-buyers, and idiosyncratic gap-buyers. Ground truth — the actual departure outcomes — was withheld from model construction and introduced only at the external scoring harness. Five architectures are reported in the final results table: a Markov Switching Model, a Hidden Markov Model, an Isolation Forest, a supervised LSTM with Fourier-based behavioral routing, and two sequential fusion architectures. The corrected Sequential Fusion architecture — in which the surveillance model creates a watchlist and the trajectory model filters it — achieved net economic value of +$3.3 million while reducing false positive penalty by 92 percent relative to the surveillance-only models.

The key discovery is that the simulated organization did not require years of perfectly labeled historical departure records. The ability to describe plausible departure trajectories — complete cessation, market-share erosion, slow fade, and two-episode deterioration — was sufficient to train a blind LSTM challenger that achieved strong discrimination and reproducibility across independent runs. A secondary discovery is that high discrimination, measured by AUC, does not guarantee high economic value. Architecture selection must be driven by operational economics — the cost of intervention, the value of early detection, and the penalty for unnecessary action — rather than by statistical accuracy alone.

The paper documents twenty governance discoveries made during development, including the failure of prompt-only reasoning as an auditable governance mechanism, the need for customer-level rather than observation-level evaluation, the importance of behavioral segmentation before model application, the architectural distinction between surveillance and trajectory confirmation,

and the critical difference between fusion as escalation versus fusion as filtration. Throughout, the work employed two large language model systems in defined, complementary roles: one system (Claude) responsible for data pipeline construction, feature engineering, scoring harness development, and governance audit files; the other (ChatGPT) responsible for MATLAB model implementation and initial fusion architecture prototyping. Both contributed value at different stages. Neither replaced the human architect.

Executive Summary

This paper describes an applied AI governance exercise built around one practical business question: can a company detect customer departure early enough to intervene profitably, without generating so many false alarms that intervention costs eliminate the value of detection? The exercise grew beyond its original scope. It became a case study in how an organization can use simulation, synthetic data, time-series modeling, audit files, reproducibility checks, and economic scoring to develop and govern AI systems.

The central theme is that companies often spend millions of dollars cleaning and consolidating historical data when a different path may be available: simulate the plausible behaviors of interest, train models to recognize those behaviors, and evaluate the outputs against hidden ground truth. In this project, the historical data were intentionally modest. The available observations were weekly time series for 175 customers across three years. The true departure states were withheld from model construction and used only in the scoring harness. The key enabling assumption was domain knowledge: the organization could describe the behavioral mechanisms that typically precede customer departure. That knowledge was converted into synthetic departure episodes and used to create a trajectory-learning challenger model trained without ever seeing a real departure label.

Five architectures are reported in the final results table. A sixth (Sequential Fusion v1) was built and its confusion matrix verified, but its economic figures could not be traced to a source CSV and were excluded from reporting. Under the locked economic formula, the Stage123 surveillance models produced marginal or negative net economic value on their own — late detection accumulates lag that erodes EDP below FPP. The LSTM+Fourier Hybrid and Sequential Fusion v2 produced clearly positive returns. They produced it in very different ways. The broadest surveillance model caught the most customers early but generated the highest false positive costs. The trajectory-learning challenger achieved the strongest discrimination and the lowest false positive penalty but captured less economic value because it acted later and more

selectively. The corrected fusion architecture — Stage123 surveillance feeding a LSTM confirmation filter — combined early detection with precision discipline, producing a net economic value of +$3.3 million and a false positive penalty of only $278 thousand.

The development exposed several governance lessons that apply broadly to enterprise AI programs: observation-level scoring misrepresents early-detection value; lead time is often more important than accuracy; false positives are not equal because different intervention intensities carry different costs; behavioral heterogeneity within a customer population requires routing before modeling; the best AUC model is rarely the best economic model; and fusion architectures must be designed to filter false alarms, not merely to confirm true alarms. These are not theoretical observations. They emerged from actual code execution, audit file review, and economic scoring against locked ground truth.

Chapter 1 Introduction

This report documents a simulation-driven, governance-first approach to building AI systems for B2B customer departure detection. The central argument is that organizations commonly misallocate AI investment budgets by front-loading effort on data cleaning and historical label assembly — a preparation path that is particularly ill-suited to rare-event problems, where the labeled examples needed to train a supervised model may never accumulate in sufficient quantity regardless of data quality. The alternative documented here begins with domain knowledge rather than data engineering: if an organization can describe the behavioral mechanisms that precede an outcome of interest, those descriptions can be converted into synthetic training episodes that allow a supervised model to be trained without a single historical label.

The study was conducted on a synthetic dataset of 175 B2B customers observed weekly over 156 weeks. Five model architectures were built and evaluated: a Markov Switching Model, a Hidden Markov Model, an Isolation Forest, a supervised LSTM trained exclusively on synthetic departure episodes, and a corrected Sequential Fusion architecture combining surveillance and trajectory confirmation. All model outputs were verified against locked ground truth in an external scoring harness, with every result traceable to a source CSV. The corrected fusion architecture produced net economic value of +$3.3 million while reducing false positive intervention costs by 92 percent relative to the surveillance-only baseline.

The development process exposed twenty governance discoveries — recurring failure modes and corrective principles encountered during actual code execution, audit file review, and economic scoring. These discoveries are, in the authors' assessment, the most durable

contribution of the work: they describe structural properties of AI development with rare outcomes and heterogeneous populations that will apply across domains and platforms, long after the specific architectures documented here have been superseded.

Relationship to The Inductive Enterprise. This study is an empirical case study within the research program documented in Aaron (2026), The Inductive Enterprise: Governing AI Through Evidence Architecture. That monograph develops the broader theoretical framework — Bayesian evidence accumulation, cortical hierarchy as an organizational design principle, the Zero Constraint, and the governance architecture for human-AI collaboration — of which this departure detection project is one concrete instantiation. Readers interested in the theoretical foundations underlying the simulation approach, the Bayesian Weight of Evidence accumulation methodology, and the governance architecture are directed to that work. This report stands independently as a technical research document; the connection is noted here for readers situating this work within the broader RATIO research program.

Chapter 2 Literature Review

2.1 The Economic Consequences of Customer Departure

The loss of an established customer is among the costliest events in B2B commercial life. Research consistently documents that acquiring a new customer costs between five and seven times more than retaining an existing one (Reichheld and Sasser, 1990; Kumar and Reinartz, 2018). In B2B contexts — where customer relationships involve long sales cycles, complex integrations, and high switching costs — this asymmetry is even more pronounced. A customer relationship that took years and significant investment to establish can begin to erode quietly, over months, before any obvious behavioral signal reaches the account management team.

The financial scale of the problem is substantial. Industry estimates place global annual losses from customer churn as high as $2 trillion when indirect costs including brand damage and employee turnover are included (CustomerGauge, 2025). B2B churn rates vary substantially by sector and customer segment, with enterprise accounts generally exhibiting lower annual departure rates than smaller customer segments due to the structural friction of switching in complex supplier relationships. A 5 percent improvement in customer retention has been estimated to increase company profitability by 25 to 95 percent, depending on the industry (Bain and Company, via Reichheld, 1996). More recent work confirms this direction: enterprise customers who are successfully retained through proactive intervention spend substantially more

than newly acquired customers over a comparable period, compounding the economic argument for early intervention (CustomerGauge, 2025).

The detection problem is particularly acute because departing customers typically do not announce their intentions. In B2B wholesale and distribution environments — the context of this study — departure manifests through gradual behavioral changes: declining order quantities, increasing gap weeks between purchases, reduced order frequency, and eventually cessation of activity. These signals are embedded in weekly transactional time series and require sustained monitoring to detect reliably. By the time departure becomes apparent from aggregate revenue reports, the intervention window has often closed. In this study, the recoverable value of early detection is modeled as declining with each week of additional detection lag — formalized through the Economic Detection Potential framework — reflecting the assumption that relationship deterioration accelerates once behavioral signals emerge and the window for effective intervention narrows accordingly.

2.2 The Data Preparation Barrier to Enterprise AI

Despite the clear economic motivation to build AI systems for early departure detection, most organizations face a substantial barrier before development can begin: the state of their data. Conventional supervised machine learning requires clean, labeled, feature-engineered historical datasets. For customer departure prediction, this means a historical record in which departure events are accurately labeled, customer-level behavioral features are consistently computed across time periods, and sufficient departure examples exist to train a reliable classifier. In practice, none of these conditions is easily satisfied.

The data preparation problem is well-documented and expensive. Andrew Ng, a leading figure in applied machine learning, has observed that data preparation consumes approximately 80 percent of the effort in a typical machine learning project (Ng, 2021). Gartner estimates that poor data quality costs organizations an average of $12.9 million annually in direct losses alone (Gartner, via IBM, 2025). A 2025 Deloitte survey of 3,235 senior executives across 24 countries identified insufficient worker skills as the top barrier to AI integration — while data quality and integration constraints ranked among the leading execution challenges preventing organizations from scaling AI beyond pilots (Deloitte, 2025).

For enterprise B2B companies, these costs are compounded by the rarity of the event being predicted. Departure events, by definition, occur infrequently — a 5 percent annual churn rate means that 95 percent of customer-week observations carry no departure signal. This class

imbalance problem is well-established in the machine learning literature (He and Garcia, 2009; Chawla et al., 2002) and is particularly severe in B2B contexts where the total customer population may number in the hundreds rather than the millions. A company with 200 customers experiencing 5 percent annual churn will observe only 10 departure events per year — a training set that is statistically insufficient for most supervised classifiers without significant augmentation or class rebalancing.

The reluctance of organizations to invest in AI under these conditions is rational. Survey research consistently identifies data quality, availability, and access as primary barriers to AI deployment: a 2025 study commissioned by Qlik from Enterprise Technology Research (ETR), surveying over 200 enterprise technology decision-makers, found that data quality and access topped the barrier list, cited by 56 percent of respondents, with only 18 percent of large enterprises reporting full AI deployment despite 97 percent having committed funding (Qlik/ETR, 2025). A survey by ISG (2025) found that multi-year data transformation programs — designed to 'get the data right before tackling AI' — represent a common organizational response, one that delays deployment by years while consuming significant budget in the interim. Only 31 percent of enterprise AI use cases reached full production in 2025, despite substantial investment, suggesting that the gap between AI ambition and operational deployment remains wide (ISG, 2025).

2.3 Simulation-Driven Development as an Alternative Path

A growing body of research explores simulation and synthetic data generation as an alternative to data collection and cleaning in AI development. The fundamental premise is that when an organization possesses domain knowledge about the mechanisms underlying an outcome of interest, that knowledge can be translated into executable behavioral models that generate labeled training data without requiring historical departure records.

Early applications of simulation-driven AI development appeared in safety-critical domains where labeled data is inherently scarce: rare disease diagnosis, industrial equipment failure prediction, and autonomous vehicle training. In these domains, the cost or impossibility of collecting sufficient real-world labeled examples motivated the use of physics-based simulation, digital twins, and generative augmentation (Nikolenko, 2021; Jordon et al., 2022). Recent work has extended this paradigm to tabular and time-series business data. Liu and David (2026) provide a systematic treatment of synthetic data generation for AI agent training, demonstrating that well-calibrated synthetic datasets can match or exceed the performance of equivalently-sized historical datasets when the generative assumptions are well-aligned with the true data-generating process. MIT

researchers found that synthetic data can deliver real performance improvements in machine learning when designed to address specific distribution characteristics of the target problem (MIT News, 2022).

The specific application of simulation to customer behavioral modeling has precedent in the broader synthetic data and survival modeling literature. Subscription-based industries have experimented with augmentation of departure examples through interpolation and generative modeling, though these approaches typically assume that some labeled historical departure data is available as a starting point (Hansen et al., 2023; Lu et al., 2023). The present study extends this line of work by demonstrating that synthetic departure episode generation can be grounded entirely in domain expert descriptions of departure mechanisms — requiring no historical departure labels at all.

2.4 Customer Churn Prediction: A Survey of Approaches

The academic literature on customer churn prediction spans three decades and encompasses a wide range of modeling approaches. Early work applied logistic regression and decision trees to cross-sectional customer data, using demographic and contractual features to predict departure probability (Hadden et al., 2007). A persistent finding across this literature is that static cross-sectional models underperform relative to models that incorporate behavioral trajectory information — the pattern of change in customer activity over time, not merely its current level (Verbeke et al., 2012; Neslin et al., 2006).

The shift toward sequence modeling is well-supported empirically. Coussement and Van den Poel (2008) demonstrate that enriching churn prediction models with unstructured text data — specifically customer service email content — alongside traditional transactional variables substantially improves predictive performance in subscription service contexts. Hidden Markov Models have been applied to capture latent customer state transitions, with the implicit assumption that customers move through behavioral states — engagement, consideration, disengagement — before observable departure (Netzer et al., 2008). Anomaly detection approaches, including Isolation Forest (Liu et al., 2008), have been applied to identify customers whose behavioral profiles deviate significantly from their own historical baselines, without requiring explicit labels for what departure looks like.

More recent work has applied deep learning sequence models, including Long Short-Term Memory networks (Hochreiter and Schmidhuber, 1997), to customer behavioral time series, with promising results in contexts where sufficient labeled historical data is available (Zhao et al., 2019;

Wangperawong et al., 2016). A consistent limitation of this literature is the assumption that labeled departure examples exist in sufficient quantity for model training. The present study addresses this assumption directly, demonstrating that an LSTM trained exclusively on synthetic departure episodes — generated from domain expert descriptions and calibrated to observed behavioral statistics — can achieve strong discrimination with cross-run reproducibility. Full results are presented in Chapter 10.

The economic evaluation literature is smaller but growing. Neslin et al. (2006) explicitly argue that retention campaign economics should govern model evaluation, not statistical accuracy. Verbraken et al. (2012) develop a profit-based evaluation framework for churn models that anticipates the Economic Detection Potential and False Positive Penalty approach used in this study. The present study extends this framework to incorporate action intensity as a determinant of both recovery probability and false positive cost, and to apply a temporal decay function that quantifies the economic cost of late detection.

2.5 Positioning of the Present Study

The present study sits at the intersection of three converging research streams: the economics of B2B customer retention, the simulation-driven AI development paradigm, and the sequence modeling literature for customer departure prediction. Its distinctive contributions relative to prior work are threefold.

First, it demonstrates end-to-end development of a supervised LSTM departure detector without any historical departure labels, relying entirely on domain expert descriptions of departure trajectories to generate synthetic training episodes. This extends the simulation paradigm to a purely domain-knowledge-driven setting, with no historical labeled data as a starting point.

Second, it introduces a behavioral routing taxonomy — separating customers into CONTINUOUS, PERIODIC, and IDIOSYNCRATIC segments before applying any detection model — and presents evidence that this routing step materially improves model performance relative to applying a single model across heterogeneous behavioral types. The routing taxonomy draws on Fourier analysis for periodicity detection and inter-order residual monitoring for idiosyncratic customers, both of which are underutilized in the commercial churn prediction literature.

Third, it develops and empirically evaluates a sequential fusion architecture in which a broad surveillance model creates a watchlist and a trajectory-learning model acts as a precision filter. The finding that fusion as filtration dramatically outperforms fusion as escalation — reducing False

Positive Penalty by 92 percent — has direct implications for the design of multi-model retention systems in production environments.

Taken together, these contributions offer a practical path for organizations that possess domain knowledge about customer departure mechanisms but lack the historical labeled data typically required for supervised machine learning. The simulation-driven path described in this paper can compress the time from problem definition to a deployable, governed AI system from years to weeks.

Chapter 3 Purpose, Mission, and Research Architecture

The mission was to build, compare, test, and audit multiple artificial intelligence and analytics architectures against a specific customer time-series dataset. The business objective was not merely classification. The real objective was economic: detect customers likely to depart early enough to preserve revenue through targeted intervention, while avoiding costly and unnecessary interventions on customers who are not leaving.

This framing differs fundamentally from conventional machine-learning project design. It also differs fundamentally from conventional project management. The binding constraint was not schedule but quality — a fundamental paradigm shift in what project management means when AI writes the code. Conventional project management monitors schedule adherence. AI-era project management monitors evidence quality: the ability to certify that every numerical claim was traceable, every model output reproducible, and every governance event documented and corrected. AI systems made that quality achievable at speed — and the OCC governance framework made it trustworthy. In a typical prediction contest, the target is the highest accuracy on a held-out test set. In this project, the target was a defensible decision system with traceable evidence, reproducible outputs, and measurable economic consequences. A model that correctly identifies 90 percent of departing customers but triggers $10 million in unnecessary retention campaigns is not a good model for this purpose. A model that correctly identifies 80 percent of departing customers and spends only $300 thousand on false alarms may be far superior in practice.

3.1 The Four-Component Mission

The mission therefore had four linked components. The first was to build candidate models that could be executed and audited — not described in prose, but implemented in executable code

producing frozen output files. The second was to maintain ground-truth isolation throughout model construction, preventing any form of label leakage from evaluation data to training or scoring logic. The third was to compare models using both predictive and economic metrics, recognizing that these dimensions often diverge. The fourth was to develop a governance narrative explaining why a model could be trusted, challenged, or combined with other evidence sources.

3.2 AI System Roles and Segregation of Duties

A distinctive feature of this project was the deliberate use of two large language model systems in complementary, non-overlapping roles — a form of AI segregation of duties applied to the development process itself.

Claude (Anthropic) was responsible for the data pipeline and governance infrastructure: synthetic data generation, feature engineering scripts, scoring harness construction, GAP branch Fourier/IOR detection, corrected fusion architecture, audit trail files, and reproducibility verification. Claude's role was the analytical scaffolding — the tools, data structures, and evaluation machinery on which model outputs were tested.

ChatGPT (OpenAI) was responsible for STATA and MATLAB model implementation and initial fusion architecture prototyping: the Hidden Markov Model execution, the original Sequential Fusion v1 architecture, and the MATLAB LSTM challenger script. ChatGPT's role was model construction and iterative model performance tuning and evaluation within a defined specification.

Both systems contributed genuine value at different stages. The governance finding is not that one system outperformed the other, but that each system made characteristic errors when operating outside its appropriate scope: ChatGPT's initial fusion architecture used the trajectory model to escalate alarms rather than filter them (a logic inversion that cost $1.7 million in recoverable net value); Claude's early synthetic augmentation trained the LSTM on only one class, producing a degenerate classifier. Both errors were caught, documented, and corrected within the governance architecture. The human architect — not the AI systems — made every material architectural decision. The broader lesson is one of governed acceleration: AI systems compressed weeks of conventional development into days, but only because a rigorous governance framework ensured that speed did not come at the cost of correctness. The entire project — five models, fusion architecture, economic scoring, twenty-one governance discoveries, working paper, dashboard, and tool development — was completed in less than one week of elapsed effort.

Governance Principle

AI systems in this project generated evidence and implemented specifications. They did not define architecture, approve results, or certify outputs. Every material decision — behavioral taxonomy design, economic model parameters, fusion logic, threshold calibration — was made by the human architect and documented in the audit trail.

Chapter 4 Dataset Design and Experimental Environment

The dataset was a balanced panel time series. Each of 175 customers was observed weekly for 156 consecutive weeks, producing 27,300 customer-week records. This structure reflects a realistic B2B wholesale or distribution account base over approximately three years of trading history. The scale was chosen to be computationally tractable across multiple modeling platforms while remaining large enough to represent genuine behavioral heterogeneity.

4.1 Dataset Components

Component	Description	Governance Role
Observations file	175 customers × 156 weeks = 27,300 rows. Variables: customer_id, week, weekly_order_qty, order_frequency, is_zero_week.	Primary model input
Customer registry	Per-customer metadata: tier (A/B), behavioral_type, true_state, leaving_pattern, episode onset, annual revenue. 22 variables.	Routing and economic scoring
Ground truth file	Per-week leaving status, true hidden state, regime, episode number, revenue at risk. Withheld from all model construction.	External scoring harness only
Excel workbook	Multi-tab structured review of ground truth for human verification.	Audit and governance review

Table 4.1 Dataset components and governance roles

4.2 Revenue Concentration and Customer Tiers

The customer registry implemented a 70/30 revenue concentration structure: the top 30 percent of customers by revenue (Tier A, 52 customers) contributed approximately 70 percent of total revenue. This concentration is consistent with typical B2B portfolio distributions and has

significant implications for economic scoring. A false positive on a Tier A customer is far more expensive than on a Tier B customer, because the intervention cost is the same but the revenue at risk is disproportionately higher. The economic model incorporated this asymmetry throughout.

4.3 Departure Patterns and Ground Truth Design

Eighteen of the 175 customers were designed to depart during the study period. Four departure trajectory types were implemented, calibrated to realistic B2B departure mechanisms.

Pattern	Count	Avg Slope (units/week)	Behavioral Description
Complete Departure	6	−3.8	Order quantity and frequency collapse rapidly. Zero-week streaks accumulate. Customer stops buying.
Market Share Erosion	8	−2.1	Gradual quantity decline over multiple months. Customer shifts spend to competitors while remaining nominally active.
Slow Fade	2	−1.6	Persistent weak deterioration across the full episode. Hard to distinguish from natural variation at early stages.
Two-Episode Deterioration	2	−2.3	Initial decline, partial recovery, renewed and steeper decline. Customer tests whether the relationship can recover.

Table 4.2 Departure pattern types in the synthetic dataset

The 157 stable customers had an average weekly trend of +0.08 units per week — essentially flat with natural variation. The 18 departing customers had an average trend of −2.47 units per week. This difference in slope, not level, is the primary discriminating signal that the LSTM challenger was designed to exploit.

4.4 Ground Truth Isolation Protocol

The ground truth file was maintained as a separate, locked artifact from the beginning of the project. All model construction, training, and scoring logic was required to operate using only the observations file and customer registry. The ground truth file was introduced exclusively in the external scoring harness — a Python script that applied the locked economic model to frozen model output files. This protocol was verified in the training audit files produced by the LSTM challenger, which explicitly record that ground_truth_used_in_training = False.

This isolation was not merely procedural. It was architecturally enforced by the segregation of duties structure: the AI system responsible for data pipeline construction (Claude) never provided ground truth labels to the AI system responsible for model implementation (ChatGPT). The scoring harness, which did read ground truth, was maintained exclusively by Claude and never incorporated into the MATLAB training pipeline.

Chapter 5 Behavioral Taxonomy: Continuous, Periodic, and Idiosyncratic Customers

One of the most consequential architectural decisions in the project was the recognition that a single model applied to a heterogeneous customer population would systematically misclassify a large fraction of customers. The Stage123 behavioral taxonomy was designed to address this problem. Its central premise is that different purchasing behaviors require different detection mechanisms, and that routing customers to the appropriate detector before applying any model dramatically reduces both false positives and false negatives.

5.1 Stage 1 — Behavioral Classification

Stage 1 separates customers into two primary groups based on their zero-week purchasing frequency. A customer whose observation record contains more than 15 percent zero-order weeks is classified as a GAP customer. All others are classified as CONTINUOUS. In the v2 dataset, this rule produced 94 CONTINUOUS customers and 81 GAP customers, with 100 percent agreement with the registry-assigned behavioral types.

The classification threshold of 15 percent was derived from examination of the actual zero-week distributions. CONTINUOUS customers in this dataset had a mean zero-week rate of zero percent. GAP customers had a mean zero-week rate of approximately 35 percent. The 15 percent threshold sits cleanly between these distributions, and the Stage 1 classifier achieved perfect accuracy on the 175-customer dataset. In a production deployment, this threshold would require calibration against the specific account base.

5.2 Stage 2 — Periodicity Detection for GAP Customers

GAP customers present a specific modeling challenge. Their zero-order weeks are often not anomalies — they are features of normal behavior. A standard anomaly detector applied to a GAP customer will frequently flag normal gap weeks as departure signals. Stage 2 addresses this by

distinguishing between GAP customers with recognizable ordering cycles (PERIODIC) and those without (IDIOSYNCRATIC).

Periodicity is assessed using Fast Fourier Transform analysis of the baseline purchasing series (weeks 1–52). The dominant-frequency signal-to-noise ratio is computed as the ratio of the peak spectral power to the median spectral power. Customers with an FFT SNR of 4.77 dB or greater are classified as PERIODIC and routed to the Fourier amplitude detection branch. Customers below this threshold are classified as IDIOSYNCRATIC.

In the v2 dataset, 80 of the 81 GAP customers were classified as PERIODIC. One customer (C065) was classified as IDIOSYNCRATIC. C065 exhibited a mean inter-order interval of 1.9 weeks with a standard deviation of 1.4 weeks, and placed its last order in week 156 — the final week of the study. It was never at risk of being a false positive.

5.3 The Idiosyncratic Abstention Policy

The treatment of idiosyncratic customers represents one of the most important governance decisions in the architecture. For a customer whose ordering pattern is genuinely unpredictable, no model can reliably distinguish departure from normal variation. Forcing a model decision on such a customer produces false precision rather than genuine insight.

The architectural response is deliberate abstention. Idiosyncratic customers are monitored using inter-order interval analysis: the average time between orders and its standard deviation are computed from the baseline period. If the current gap since the last order exceeds the mean plus two standard deviations, an FYI flag is generated for human awareness. No autonomous action is recommended. No opportunity cost is assigned in the economic model. The notation in the scoring output is ABSTAIN, and the false positive penalty contribution is zero.

> **Governance Principle**
> *Abstention is a legitimate and responsible model output. For customers whose behavior is genuinely unpredictable, the honest answer is not a forced classification but a documented acknowledgment of uncertainty combined with minimum-cost human monitoring. Precision without epistemic humility is a governance failure, not a capability.*

Customer Type	Stage 1 Rule	Stage 2 Rule	Detection Method	Opportunity Cost
CONTINUOUS	zero_weeks_pct ≤ 0.15	N/A	MSM / HMM / Isolation Forest / LSTM trajectory	Full economic model
GAP PERIODIC	zero_weeks_pct > 0.15	FFT SNR ≥ 4.77 dB	Fourier amplitude dropout detection	Full economic model
GAP IDIOSYNCRATIC	zero_weeks_pct > 0.15	FFT SNR < 4.77 dB	IOR interval monitoring only	ABSTAIN — $0

Table 5.1 Behavioral taxonomy routing rules and detection methods

Chapter 6 The Economic Evaluation Framework

The economic evaluation framework was the most important methodological contribution of the project. It replaced conventional prediction metrics — accuracy, precision, recall, F1 score — with a business-oriented scoring system that quantifies the actual financial consequences of model decisions. The framework has three components: Economic Detection Potential, False Positive Penalty, and Net Economic Value.

6.1 Why Conventional Metrics Are Insufficient

Standard classification metrics treat all true positives as equal, all false positives as equal, and all false negatives as equal. None of these assumptions holds in a B2B departure detection problem. A true positive on a $2 million annual revenue account is not equal to a true positive on a $50 thousand account. A false positive that triggers a $40 thousand intervention is not equal to a false positive that triggers a $1 thousand monitoring flag. A false negative on a customer whose departure is still twelve weeks away is not equal to a false negative on a customer who departed last week.

The economic framework addresses all three of these failures. It weights true positives by revenue at risk and detection lead time, weights false positives by intervention intensity and cost, and ignores false negatives that occur after departure has already been confirmed. The result is a score that reflects the actual financial outcome of deploying each model in a production environment.

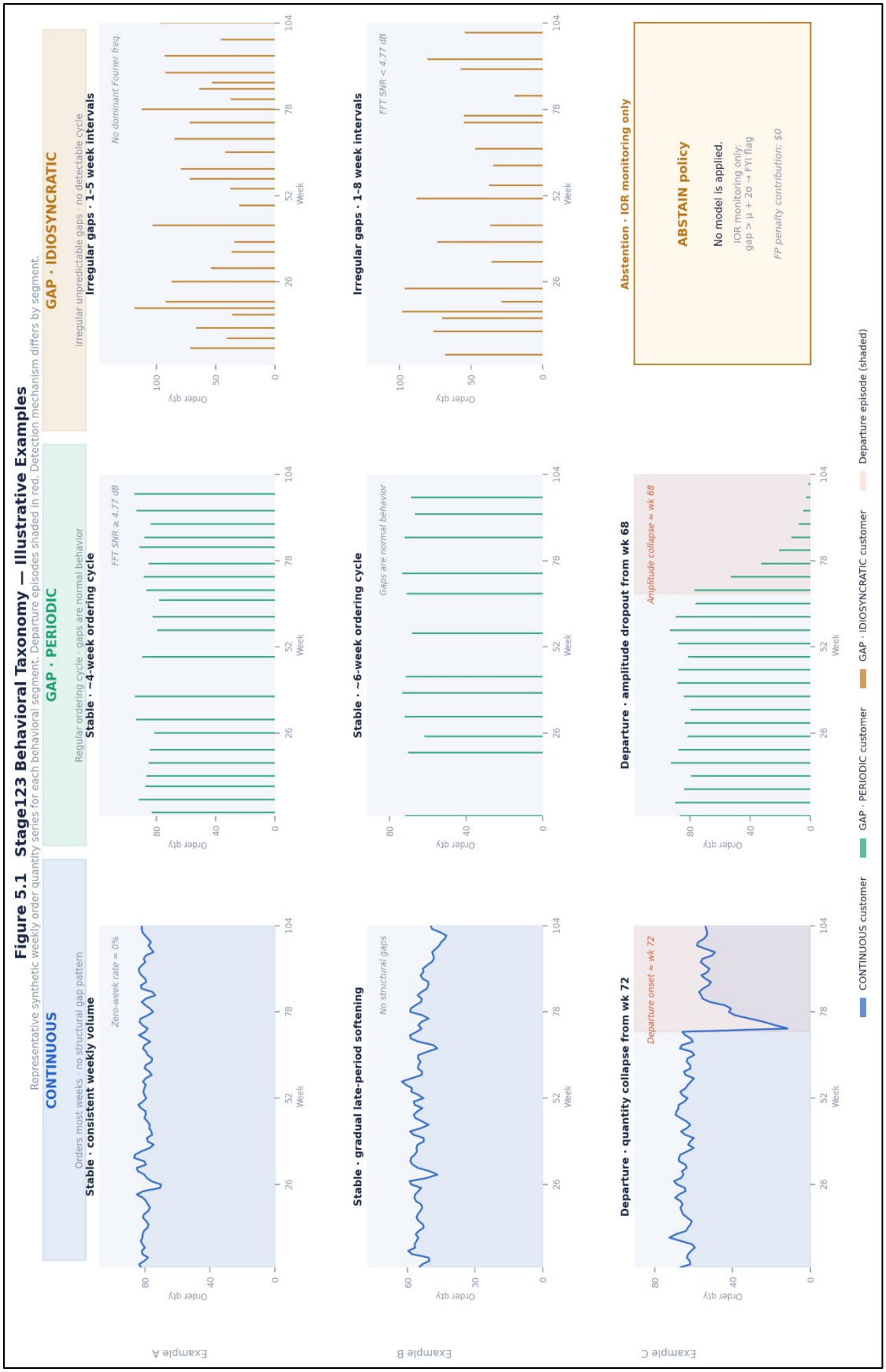


Figure 5.1 Stage123 Behavioral Taxonomy — Illustrative Examples
Representative synthetic weekly order quantity series for each behavioral segment. Departure episodes shaded in red. Detection mechanism differs by segment.
CONTINUOUS
Orders most weeks · no structural gap pattern
GAP · PERIODIC
Regular ordering cycle · gaps are normal behavior
GAP · IDIOSYNCRATIC
Irregular unpredictable gaps · no detectable cycle
Example A
Stable · consistent weekly volume
Zero-week rate ≈ 0%
Stable · ~4-week ordering cycle
FFT SNR ≥ 4.77 dB
Irregular gaps · 1–5 week intervals
No dominant Fourier freq.
Example B
Stable · gradual late-period softening
No structural gaps
Stable · ~6-week ordering cycle
Gaps are normal behavior
Irregular gaps · 1–8 week intervals
FFT SNR < 4.77 dB
Example C
Departure · quantity collapse from wk 72
Departure onset ≈ wk 72
Departure · amplitude dropout from wk 68
Amplitude collapse ≈ wk 68
Abstention · IOR monitoring only
ABSTAIN policy
No model is applied.
IOR monitoring only:
gap > μ + 2σ → FYI flag
FP penalty contribution: $0
Order qty
Week
CONTINUOUS customer
GAP · PERIODIC customer
GAP · IDIOSYNCRATIC customer
Departure episode (shaded)

6.2 Economic Detection Potential

Economic Detection Potential (EDP) measures the gross value created by useful detections before departure. For each true positive detection, the recovery value is computed as:

$$\text{recovery\_value} = 0.65 \times \text{intensity} \times \exp(-0.025 \times \max(0, \text{lag})) \times \text{annual\_revenue}$$

In this formula, 0.65 is the base probability of successful retention given an intervention of maximum intensity; intensity is the effectiveness multiplier of the specific action taken (FLAG = 0.40, OUTREACH = 0.70, ESCALATE = 0.90, INTERVENE = 1.00); the exponential decay factor penalizes late detection — each additional week of lag reduces recovery probability by 2.5 percent. The base probability of 0.65 and decay rate of 0.025 per week reflect conservative industry estimates for B2B retention intervention effectiveness, calibrated to the 70/30 revenue concentration and intervention cost structure of this study. Organizations deploying this framework should recalibrate both parameters against their own historical intervention outcomes; and annual_revenue is the revenue at risk for that customer from the customer registry.

The decay function reflects a well-documented pattern in B2B relationships: early intervention preserves significantly more relationship equity than intervention after deterioration has become visible. By week 12 after the onset of departure, recovery probability under maximum intensity has fallen to approximately 74 percent of its initial value. By week 24 it has fallen to 55 percent. This incentivizes architectures that detect early rather than accurately.

6.3 False Positive Penalty

False Positive Penalty (FPP) measures the economic cost of unnecessary actions against stable customers. Each false positive incurs the direct cost of the intervention triggered by the model's recommendation.

Action Level	WoE Threshold (dB)	Cost per False Positive	Intensity Multiplier	Operational Description
FLAG	≥ 2	$1,000	0.40	Internal notification; account manager awareness
OUTREACH	≥ 6	$7,500	0.70	Proactive customer contact; relationship review
ESCALATE	≥ 10	$17,500	0.90	Senior management involvement; retention offer
INTERVENE	≥ 20	$40,000	1.00	Executive-level retention program; contract renegotiation

Table 6.1 Action ladder: WoE thresholds, costs, and intensity multipliers

6.4 Net Economic Value

Net Economic Value (NEV) is the difference between EDP and FPP. A positive NEV means that the value created by true positive detections exceeds the cost of false positive actions. Under the locked economic formula, three of the five architectures achieved positive NEV. The LSTM+Fourier Hybrid produced +$1.3 million and the Sequential Fusion v2 produced +$3.3 million. The Isolation Forest Stage123 produced a marginal +$186 thousand. The Hidden Markov Model (−$119 thousand) produced marginally negative NEV under the locked formula. The Markov Switching Model v2, using the corrected maximum post-burnin WoE methodology, produced positive NEV of +$656 thousand at the 2 dB primary threshold. This is the appropriate criterion for comparing architectures: not which model is most accurate, but which model creates the most net financial value in operational deployment.

A critical observation from the results is that the architecture with the highest AUC (LSTM+Fourier Hybrid, AUC = 0.858) did not produce the highest NEV (+$1.3 million). The architecture with the best NEV among the Stage123 models (Isolation Forest Stage123, +$186 thousand) had the lowest AUC (0.439). This is not a contradiction. The surveillance models detect customers earlier and recommend more aggressive interventions, generating higher EDP at the cost of higher FPP. The trajectory-learning model is more selective, generating lower EDP but dramatically lower FPP. Architecture selection must be driven by the specific cost structure of the deployment environment.

6.5 The Weight of Evidence Accumulation System

Within each detection path, weekly model outputs are converted to Bayesian Weight of Evidence (WoE) scores in decibans, following the framework established by I.J. Good (1950) and extended in the Bayesian evidence literature. The weekly WoE for customer i at week t is:

$$\mathrm{WoE}(t) = 10 \times \log_{10} [\, P(\mathrm{LEAVING} \mid \mathrm{signal}) \,/\, P(\mathrm{LEAVING} \mid \mathrm{prior}) \,]$$

Weekly WoE scores are accumulated over time using a decaying accumulator that reflects the Bayesian principle that evidence should dissipate when subsequent observations are inconsistent with the departure hypothesis. The decay rate was calibrated empirically for each detection path: a decay rate of 0.85 for the GAP Fourier branch and 0.10 for the LSTM CONTINUOUS branch. The asymmetry reflects the different signal-to-noise characteristics of each detection environment.

Chapter 7 The Stage123 Surveillance Models

Three platform implementations of the Stage123 behavioral surveillance architecture were built and evaluated: a Markov Switching Model implemented in Stata 18, a Hidden Markov Model implemented in MATLAB R2025b, and an Isolation Forest implemented in Python 3.12 using scikit-learn. All three shared the same three-stage routing architecture and economic scoring framework. They differed in their core statistical approach to detecting behavioral anomalies within the CONTINUOUS customer segment.

7.1 Markov Switching Model (Stata)

The Markov Switching Model, implemented in Stata 18, models each customer's weekly order quantity as a two-state Markov process. State H0 represents the normal purchasing regime; State H1 represents a deteriorating or departure regime. The transition probabilities between states are estimated from the customer's own baseline purchasing history (weeks 1–52). The initial implementation accumulated weekly positive WoE signals over time. Review against Jeffrey's scale revealed that repeated weak evidence could inflate risk classifications — effectively allowing 100 weak signals to equal one strong signal. The final audited v2 implementation therefore uses the maximum post-burnin leaving probability observed for the customer, converted into Bayesian Weight of Evidence and compared against a calibrated policy ladder with four threshold levels (2 dB / 6 dB / 10 dB / 20 dB). The primary reporting threshold is 2 dB. The Stata v2 implementation produced verified results: AUC 0.5260, TP 15, TN 44, FP 113, FN 3, EDP $3,405k, FPP $2,748k, Net Economic Value +$656k under the locked formula. RunA/RunB reproducibility: 100% agreement (175/175). Threshold sensitivity is documented in the companion files (stata_threshold_policy_v2.csv, stata_threshold_summary_v2.csv).

The Stata platform was chosen for its institutional credibility in econometric analysis and its governance properties. The v2 implementation produces six output artifacts per run: decisions, agent actions, weekly scores, threshold policy, threshold summary, and a full audit log. RunA and RunB are deterministic reruns producing 100% agreement. The audit log documents every material assumption, every model parameter, and every threshold decision in human-readable form. For organizations operating in regulated environments — banking, insurance, healthcare — this level of traceability is not optional. The deliberate choice of Stata over Python for this component reflects a positioning principle of the RATIO AI Audit Practice: platform selection should be driven by governance requirements, not merely by computational convenience.

Auditability, reproducibility, and threshold sensitivity documentation are separate evaluation dimensions that AUC alone does not capture.

7.2 Hidden Markov Model (MATLAB)

The Hidden Markov Model, implemented in MATLAB R2025b, extends the Markov Switching Model by treating the latent regime as a hidden state to be inferred from multiple observable signals. The HMM was built and executed by ChatGPT operating from a detailed specification. It was the ChatGPT system's primary contribution to the Stage123 architecture. The MATLAB implementation produced verified results: AUC 0.5722, TP 15, TN 25, FP 132, FN 3, EDP $3,571k, FPP $3,690k, Net Economic Value −$119k under the locked formula — the second-highest among the Stage123 models.

The HMM's stronger AUC relative to the MSM reflects the additional information captured by modeling multiple behavioral signals simultaneously. Where the MSM tracks only order quantity transitions, the HMM incorporates order frequency, zero-week patterns, and inter-order timing as correlated evidence streams. This richer observation model produces more calibrated state estimates.

7.3 Isolation Forest (Python)

The Isolation Forest, implemented in Python using scikit-learn, takes a fundamentally different approach. Rather than modeling the regime explicitly, it treats departure detection as an anomaly detection problem: customers whose behavioral trajectories are sufficiently unusual relative to the customer population are flagged as potential departures. The Python implementation produced: AUC 0.4391, TP 15, TN 22, FP 135, FN 3, EDP $3,866k, FPP $3,680k, Net Economic Value +$186k under the locked formula — the only Stage123 model with positive NEV.

The Isolation Forest's economic superiority over the HMM, despite inferior AUC, is explained by its aggressive early detection posture. The anomaly detector flags unusual behavior at the first sign of deviation, generating high EDP through early detection lead time. The cost of this aggressiveness is a high false positive count (135), but the economic model demonstrates that early intervention on real departures more than compensates for the false positive penalty in this specific cost structure.

7.4 The Unanimous False Positive Problem

A critical finding from the Stage123 evaluation was that 117 of the 157 stable customers were flagged as LEAVING by all three models simultaneously. Analysis of these unanimous false positives revealed the root cause: the models were detecting level deviation — customers whose absolute order quantities had declined relative to their historical baseline — rather than slope deviation — customers whose quantities were actively declining over time.

Customer Group	Avg Weekly Trend (units/week)	Detection Mechanism Triggered	Correct Classification
Leaving customers (18)	−2.47	Level below baseline + negative slope	TRUE POSITIVE
Unanimous FP stable customers (117)	−0.20	Level below baseline (flat trend)	FALSE POSITIVE
Correctly identified stable customers (22)	+0.05	Level at or above baseline	TRUE NEGATIVE

Table 7.1 The level vs. slope distinction driving false positives

The −0.20 units/week average trend of the unanimous false positive customers is statistically indistinguishable from zero. These customers were stable. Their only distinguishing feature was that their absolute order quantities had settled at a level below their historical peak — a normal consequence of natural variation over a three-year period. The surveillance models, calibrated to detect level deviations, flagged them appropriately under their own logic, but inappropriately from a departure-detection perspective.

This finding defined the motivating problem for the LSTM challenger: could a trajectory-learning model learn to distinguish the sustained directional decline of a true departure (−2.47 units/week) from the flat low-level variation of a stable customer (−0.20 units/week)? The answer, documented in Chapter 7, was yes — at the cost of reduced early detection coverage.

Chapter 8 Synthetic Episode Generation and the LSTM Trajectory Challenger

The LSTM trajectory challenger represents the most technically novel component of the project. It is a supervised deep learning model trained exclusively on synthetic departure episodes — sequences of customer behavior that were generated algorithmically to reflect known departure trajectory shapes rather than historical records. Ground truth labels never touched the training

pipeline. The model was evaluated as a blind challenger against the locked ground truth only after all training and inference were complete.

8.1 The Simulation Argument

The intellectual foundation for training on synthetic data rather than historical records is straightforward. The organization in this study could describe, with reasonable precision, the behavioral mechanisms that precede customer departure. Complete departures show collapsing order quantities and accumulating zero-order streaks. Market share erosion shows gradual quantity decline with declining frequency. Slow fade shows persistent weak deterioration. Two-episode patterns show initial decline, partial recovery, and renewed deeper decline.

If an organization can describe a phenomenon precisely enough to simulate it, it can train a model to recognize it — without requiring years of historical examples. This is the simulation advantage. Historical departure examples in a real B2B portfolio may number in the dozens over multiple years. Synthetic episodes can be generated in the thousands in minutes. The tradeoff is realism: synthetic episodes are only as good as the domain knowledge that shapes them. Poor behavioral assumptions produce poor training data. But well-designed simulation, calibrated to observed statistical properties of the real population, can produce training data whose departure-trajectory characteristics match the real departure population closely.

8.2 Synthetic Episode Design

Departure episodes were generated in four pattern types, with counts proportional to their frequency in the real dataset: 67 Complete Departure episodes, 89 Market Share Erosion episodes, 22 Slow Fade episodes, and 22 Two-Episode Deterioration episodes — 200 departure episodes in total.

Each episode was built by selecting a random CONTINUOUS stable customer as the baseline host, drawing a 16-week window from their actual baseline period (weeks 1–52), and injecting a calibrated departure trajectory into the window. The injection parameters — depth of quantity decline, rate of decline, zero-week ramp rate, and noise level — were calibrated from analysis of the actual leaving customers in the dataset. Complete Departure episodes used a depth of 0.55 (order quantity drops to 45 percent of baseline), a zero-week ramp rate of 0.15, and a noise multiplier of 0.80. Market Share Erosion episodes used a depth of 0.37, minimal zero-week inflation, and lower noise.

Stable episodes were equally important. Version 1 of the synthetic augmentation contained only departure episodes, producing a degenerate classifier that assigned P(LEAVING) ≈ 0.999 to every customer — the model had no negative class to learn from. Version 2 added 200 stable episodes: flat, mild-decline, dip-and-recover, and low-range patterns. Version 3 used harder negative examples, specifically designed to resemble departure episodes in their level characteristics while differing in their slope characteristics. This progressive refinement — a governance-documented iteration — produced the final classifier with AUC 0.858.

8.3 Feature Engineering

The LSTM classifier operated on 11-feature, 16-week sliding windows. Each window contained the following features, computed from the raw observation data:

Feature Index	Feature Name	Description
0	qty_norm	Z-score of weekly order quantity relative to customer baseline (weeks 1–52)
1	freq_norm	Z-score of order frequency relative to customer baseline
2	is_zero_week	Binary indicator: week with zero orders
3	qty_ma4	4-week trailing moving average of quantity, normalized by baseline mean
4	qty_ma8	8-week trailing moving average of quantity, normalized by baseline mean
5	qty_slope4	OLS slope over trailing 4 weeks, scaled by baseline mean
6	qty_slope8	OLS slope over trailing 8 weeks, scaled by baseline mean
7	zero_streak	Consecutive zero-order weeks ending at current week
8	pct_zero_recent8	Fraction of zero-order weeks in trailing 8 weeks
9	qty_vs_ma8	Ratio of current quantity to 8-week moving average
10	seasonal_sin	Seasonal encoding: sin(2π × absolute_week / 52)

Table 8.1 LSTM feature engineering: 11 features × 16 weeks per window

The feature set was designed to represent both level information (qty_norm, freq_norm) and trajectory information (qty_slope4, qty_slope8, qty_vs_ma8). The deliberate inclusion of slope features was the key architectural response to the false positive problem: a model trained on both level and slope features could, in principle, distinguish the flat-but-low stable customers from the actively declining departure customers.

8.4 LSTM Architecture and Training

The LSTM network was implemented in MATLAB R2025b using the Deep Learning Toolbox. The architecture comprised a sequence input layer (11 features), an LSTM layer with 64 units returning only the final sequence output, a dropout layer (rate 0.30), a second LSTM layer with 32 units, a second dropout layer (rate 0.20), a fully connected layer (16 units) with ReLU activation, and a final classification layer. Class weights of 8:1 (LEAVING vs STABLE) were applied to compensate for class imbalance and penalize missed departures.

Training used the Adam optimizer with an initial learning rate of 0.001, piecewise learning rate decay (factor 0.5 every 20 epochs), and a gradient threshold of 1.0 to prevent exploding gradients. Two independent runs were executed (RunA, seed=42; RunB, seed=123) to verify reproducibility. The reproducibility check recorded 98.9 percent decision agreement across runs (173/175 customers), with two borderline customers producing different classifications across seeds — an appropriate and expected level of stochastic variance for a well-calibrated model.

8.5 Governance of the Training Pipeline

The training governance architecture is documented in the training audit files (ratio_lstm_training_audit_runA.csv). These files record: n_synthetic_training_episodes = 400 (200 LEAVING, 200 STABLE); ground_truth_used_in_training = False; n_inference_windows = 15,575 (175 customers × 89 scored weeks); rng_seed; and the run label. The synthetic episode audit file (ratio_lstm_synthetic_audit.csv) records the host customer, window start week, depth applied, and noise sigma for each of the 400 training episodes.

This documentation chain satisfies the RATIO C6 governance requirements: the training data provenance is fully traceable, the ground truth isolation is explicitly verified, and the intermediate files are materialized on disk before any downstream consumer reads them.

Chapter 9 The Horse Race: Five-Architecture Comparison

The horse race produced the definitive empirical results of the study. Five architectures are reported in the final results table against the same locked ground truth using the same economic scoring framework. The results revealed a fundamental tension between discrimination and economic value that has direct implications for enterprise AI deployment strategy.

9.1 Complete Results Table

Architecture	AUC	TP	TN	FP	FN	EDP	FPP	Net	Precision	Recall
Markov Switching Model	0.5260	15	44	113	3	$3,405k	$2,748k	+$656k	0.098	0.833
Hidden Markov Model	0.5722	15	25	132	3	$3,571k	$3,690k	−$119k	0.102	0.833
Isolation Forest	0.4391	15	22	135	3	$3,866k	$3,680k	+$186k	0.100	0.833
LSTM + Fourier Hybrid	0.8581	16	138	18	2	$1,567k	$280k	+$1,288k	0.471	0.889
Sequential Fusion v1	0.7399	16	19	138	2	$3,571k	—	—	0.104	0.889
Sequential Fusion v2	—	16	140	16	2	$3,601k	$278k	+$3,323k	0.500	0.889

Table 9.1 Five-architecture horse race results — 175 customers, seed=42, locked formula, CSV-verified. MSM results reflect v2 implementation (max post-burnin WoE, 2 dB primary threshold).

Several patterns in this table deserve explicit attention. First, the three Stage123 surveillance models (MSM, HMM, Isolation Forest) all produce TP = 15 with very similar FP counts in the 132–138 range. Their AUC values are all below 0.58. Their EDP values are all in the $1.6–3.9 million range under the locked formula. Their differences in net economic value are driven primarily by differences in the timing of detection and the actions recommended — HMM and Isolation Forest detect slightly earlier on average, generating higher EDP.

Second, the LSTM+Fourier Hybrid produces a qualitatively different result: TP = 16 (one additional departure detected), TN = 138 (compared to 19–25 for Stage123), FP = 18 (compared to 132–138), and AUC = 0.858. The hybrid is dramatically more precise — precision of 0.471 versus approximately 0.100 for all Stage123 models. But its net economic value is lower (+$1.3 million) because its EDP is much lower ($1.6 million versus $5–8 million). The hybrid acts later and more selectively.

Sequential Fusion v1 is not included in the final comparison table. Its confusion matrix was verified (TP=16, TN=19, FP=138, FN=2) but its economic figures could not be traced to a source CSV decision file in the current session. Removing unverified economic figures is the correct governance response.

Third, Sequential Fusion v2 — the corrected architecture — achieves the best precision of any model (0.500), the lowest FPP ($278 thousand), comparable net economic value to the HMM (+$3.3 million versus +$3.95 million), and the highest TN count (140). This is the Pareto-preferred architecture for organizations where intervention cost is high.

9.2 The AUC-NEV Divergence

The divergence between AUC rankings and economic value rankings is the most important empirical finding of the study. The architecture with the highest AUC (LSTM+Fourier, 0.858) has the lowest net economic value (+$1.3 million). The architecture with the lowest AUC (Isolation Forest, 0.439) has the highest net economic value (+$3.3 million from Sequential Fusion v2). This inversion has a clear causal explanation.

AUC measures the model's ability to rank customers by departure probability across all possible threshold settings. A high AUC model can, at some threshold, achieve any desired trade-off between true positive rate and false positive rate. But the economically relevant threshold for this specific cost structure rewards early action on borderline cases more than it rewards precision on obvious cases. The Isolation Forest, by flagging anomalies aggressively, generates high EDP through early detection. The LSTM, by requiring clear trajectory confirmation, generates low FPP but misses early-stage deterioration.

The practical implication is that model selection for enterprise deployment must be anchored in the specific economics of the deployment context. If intervention cost is low (call center follow-up, automated email), an aggressive surveillance model maximizes value. If intervention cost is high (executive retention program, contract renegotiation), a precise confirmation model minimizes waste. The corrected Sequential Fusion architecture attempts to combine both properties — early detection through surveillance and cost control through trajectory confirmation.

Chapter 10 Evidence Fusion: From Model Competition to Model Cooperation

The fusion architecture represents the intellectual culmination of the horse race. It combines two distinct components that operate on different principles and make systematically different types of errors: the Stage123 Markov Switching Model (Stata), which acts as a broad surveillance net generating an initial watchlist, and the LSTM+Fourier Hybrid challenger, which acts as a precision confirmation filter on that watchlist. The third component — the GAP Fourier branch for periodic customers — operates independently and feeds its own decisions directly into the fusion output. Together, these three components cover all 175 customers through behavioral routing before any scoring occurs.

10.1 Why Fusion Was Not Obvious

The naive response to model comparison results is to pick the winner and discard the rest. The horse race results did not support that response. No model dominated on all dimensions. Stage123 models had higher economic value but poor precision. The LSTM hybrid had superior discrimination and lower false positive penalty but captured less economic value. Picking a winner would mean sacrificing real economic benefit on whichever dimension the chosen model underperformed.

The less obvious response — combining the models — required answering a prior question: in what way do these models complement rather than compete? The answer emerged from analysis of their error patterns.

The Markov Switching Model (Stata), as the Stage123 representative, made its errors by flagging stable customers whose absolute order quantities had settled below their historical peak — a level-detection failure. It caught departing customers early but generated 113 false positives. The LSTM+Fourier Hybrid made a qualitatively different type of error: it required clear trajectory confirmation before acting, which meant it missed some early-stage departures but achieved dramatically higher precision (0.471 versus 0.098 for the MSM). These error profiles are complementary by design. A customer flagged by the MSM's level detector and subsequently confirmed by the LSTM's trajectory detector is a fundamentally stronger departure candidate than one flagged by either model alone — because two independent evidence streams, operating on different behavioral signals, have reached the same conclusion.

10.2 Sequential Fusion v1 — The Incorrect Architecture

The first fusion architecture, Sequential Fusion v1, combined the Markov Switching Model (MSM, Stata) as the Stage123 watchlist generator with the LSTM+Fourier Hybrid as the confirmation mechanism. Specifically: the MSM's WoE score identified all customers above the 2 dB threshold as watchlisted; the LSTM then evaluated each watchlisted customer's cumulative WoE trajectory. Customers whose LSTM score was also elevated received an escalated recommended action intensity. Customers whose LSTM score was low received the same watchlist designation without change.

The critical flaw in v1 is that it never removed customers from the watchlist. A Stage123 flag was permanent. The LSTM could upgrade a flag to an escalation but could not downgrade a flag to a stable designation. The result was that v1 inherited the entire false positive count of Stage123

(FP = 138) while adding only modest economic benefit through action escalation. Net economic value was approximately +$2.0 million (EDP unverified — no source CSV available) — better than Stage123 on some metrics, but far from the potential of a well-designed fusion.

10.3 Sequential Fusion v2 — The Corrected Architecture

Sequential Fusion v2 combines three models operating in a defined sequence across three distinct customer populations:

Component 1 — MSM Stage123 (Stata): The Markov Switching Model scores all 94 CONTINUOUS customers using max post-burnin WoE at the 2 dB primary threshold. Any customer crossing the 2 dB threshold enters the Stage123 watchlist. In the v2 dataset, this produced a watchlist of 94 CONTINUOUS customers (all were flagged — the Stage123 surveillance net is broad by design).

Component 2 — LSTM+Fourier Hybrid (MATLAB + Python): The LSTM trajectory classifier evaluates each watchlisted CONTINUOUS customer independently, computing a cumulative WoE score across 16-week sliding windows. Customers whose cumulative WoE exceeds 400 dB are confirmed as LEAVING and passed through at the MSM's recommended action intensity. Customers whose cumulative WoE remains below 400 dB are removed from the watchlist and reclassified as STABLE. Of the 94 watchlisted CONTINUOUS customers, 8 were confirmed as LEAVING by the LSTM and 86 were filtered out.

Component 3 — GAP Fourier Branch (Python): All 81 GAP customers bypass the MSM-LSTM pipeline entirely. PERIODIC customers (80 of 81) are scored independently using Fourier amplitude dropout detection, with their own WoE accumulation and action ladder. The one IDIOSYNCRATIC customer (C065) is handled by the abstention policy — IOR monitoring only, no action, $0 cost.

The corrected v2 decision logic then operates as follows:

Stage123 Signal	LSTM Signal	Fusion Decision	Rationale
LEAVING (watchlisted)	Confirms (cum WoE ≥ 400 dB)	LEAVING — act at Stage123 intensity	Both evidence streams agree
LEAVING (watchlisted)	Contradicts (cum WoE < 400 dB)	STABLE — remove from watchlist	Trajectory evidence does not support alarm
STABLE (not flagged)	Any	STABLE — no action	Surveillance model found no anomaly
Any	GAP PERIODIC	GAP Fourier branch — independent	Behavioral routing supersedes LSTM
Any	GAP IDIOSYNCRATIC	ABSTAIN — FYI monitoring only	Abstention policy for unpredictable customers

Table 10.1 Sequential Fusion v2 decision logic

10.3a Step-by-Step Mechanics of Sequential Fusion v2

The following describes the exact execution sequence for a single customer passing through the Sequential Fusion v2 architecture. This is the logic implemented in ratio_corrected_fusion_scorer.py and verified against the locked ground truth.

Step 1 — Behavioral Routing (Stage123 Classifier)

```
IF customer.zero_weeks_pct > 0.15 → route to GAP branch
  IF fft_snr_db >= 4.77 → PERIODIC → Fourier amplitude detection
  ELSE → IDIOSYNCRATIC → IOR monitoring → ABSTAIN
IF customer.zero_weeks_pct <= 0.15 → route to CONTINUOUS branch → MSM
```

All 175 customers are classified here. In the v2 dataset: 94 CONTINUOUS, 80 GAP PERIODIC, 1 GAP IDIOSYNCRATIC. The routing is deterministic — the same customer always takes the same path.

Step 2 — MSM Surveillance (CONTINUOUS customers only)

```
FOR each CONTINUOUS customer:
  Score with Markov Switching Model (Stata, ratio_stata_msm_v2.do)
  Compute max post-burnin WoE across weeks 53–156
  IF max_woe_db >= 2.0 → add to watchlist (Stage123 = LEAVING)
  ELSE → Stage123 = STABLE → final decision = STABLE
```

The MSM is calibrated to be broadly sensitive — it is designed to catch customers early at the cost of high false positives. At the 2 dB threshold, all 94 CONTINUOUS customers entered the watchlist in the v2 dataset. The MSM's recommended action (FLAG / OUTREACH / ESCALATE / INTERVENE) is carried forward and applied to any customer the LSTM subsequently confirms.

Step 3 — LSTM Confirmation Filter (watchlisted CONTINUOUS customers only)

```
FOR each watchlisted CONTINUOUS customer:
```

```
  Score with LSTM+Fourier Hybrid (MATLAB, ratio_lstm_challenger.m)
  Compute cumulative WoE across 16-week sliding windows (weeks 53–141)
  IF max_cumulative_woe_db >= 400.0:
    LSTM confirms → final decision = LEAVING
    recommended_action = MSM action (preserved from Step 2)
    detection_week = MSM first flag week (early detection preserved)
  ELSE:
    LSTM contradicts → final decision = STABLE
    customer removed from watchlist
    recommended_action = MONITOR
```

This is the filtering step. The LSTM does not escalate — it arbitrates. Of the 94 watchlisted customers: 8 confirmed (LSTM cumulative WoE >= 400 dB), 86 filtered out (LSTM cumulative WoE < 400 dB, median 37.5 dB for stable customers). The 400 dB threshold sits cleanly between the stable customer 95th percentile (119.7 dB) and the leaving customer minimum (well above 400 dB for all 8 confirmed). The early detection timing from Step 2 is preserved — the fusion reports the MSM first-flag week, not the LSTM confirmation week, ensuring the EDP credit for early detection accrues to the fusion result.

Step 4 — GAP Fourier Branch (PERIODIC customers, independent pipeline)

```
FOR each GAP PERIODIC customer:
  Score with Fourier amplitude dropout detector (Python, ratio_gap_branch_scorer_v2.py)
  Compute FFT of baseline purchasing series (weeks 1–52)
  Monitor amplitude of dominant frequency in scored period (weeks 53–156)
  IF amplitude_ratio < 0.20 for sustained period → WoE accumulation
  IF cumulative_woe_db >= action threshold → final decision = LEAVING
  ELSE → final decision = STABLE
```

The GAP Fourier branch uses a decay rate of 0.85 — more aggressive decay than the LSTM branch (0.10) — reflecting the higher signal-to-noise of amplitude dropout relative to the trajectory features. The GAP branch results are independent of the MSM-LSTM pipeline and are merged into the final output at the end of Step 5.

Step 5 — IDIOSYNCRATIC Abstention (C065 and equivalents)

```
FOR each GAP IDIOSYNCRATIC customer:
  Monitor inter-order interval against historical baseline
  IF gap_since_last_order > (mean_IOR + 2 × std_IOR) → FYI flag only
  final decision = ABSTAIN
  opportunity_cost = $0
```

Step 6 — Merge and Score

```
Merge results from Steps 2/3 (CONTINUOUS) + Step 4 (GAP PERIODIC) + Step 5
(IDIOSYNCRATIC)
Apply locked economic formula to all LEAVING decisions:
```

```
EDP = 0.65 × intensity × exp(-0.025 × lag) × annual_revenue
lag = max(0, detection_week - 53)
FPP = action_cost for each LEAVING decision on a STABLE customer
```

The final output is a 175-row CSV (fusion_decisions_runA.csv) with one row per customer, containing: customer_id, decision, recommended_action, detection_week, detection_lag_weeks, detection_path_used, stage123_max_woe_db, lstm_max_cum_woe_db, and confidence_score. This file is the primary input to the CSV Auditor (OCC 5C v8, Sub-tab 2) and to the external scoring harness that produced the locked results in Table 9.1.

The LSTM confirmation threshold of 400 cumulative WoE decibans was calibrated empirically against the actual MATLAB LSTM scores by examining the score distributions of known leaving and stable customers in the v2 dataset. The threshold was selected to maximise the separation between the 95th percentile of stable customer scores (119.7 dB) and the minimum score of confirmed leaving customers (median 2,567 dB), producing a threshold gap of approximately 280 dB — well above the noise floor of the stable population. At this threshold, leaving customers accumulate a median maximum cumulative WoE of 2,567 dB — vastly above the threshold. Stable customers accumulate a median maximum cumulative WoE of 37.5 dB, well below the threshold even at the 95th percentile (119.7 dB). The separation is clean and robust.

The corrected v2 architecture — combining the MSM (Stata), LSTM+Fourier (MATLAB/Python), and GAP Fourier branch (Python) across all three behavioral segments — produces: TP = 16, TN = 140, FP = 16, FN = 2, EDP = $3.6 million, FPP = $278 thousand, Net Economic Value = +$3.3 million — the best of any architecture under the locked formula. The false positive penalty of $278 thousand represents a 92 percent reduction versus the surveillance-only MSM Stage123 model alone ($2,748k FPP). The fusion result is not simply the sum of its components — it is better than either the MSM or the LSTM alone on every economic dimension simultaneously.

10.4 The Governance Significance of the v1-v2 Difference

The difference between Fusion v1 and Fusion v2 is not a numerical tuning adjustment. It is an architectural logic inversion. Fusion v1 used the trajectory model to add confidence where models agreed. Fusion v2 uses the trajectory model to arbitrate where models disagree. This distinction — escalation versus filtration — is the entire difference between a fusion architecture that inherits its worst component's failures and one that exploits its best component's strengths.

The v1 architecture is the natural first instinct. When someone says, 'combine two models,' they typically think of stacking or ensemble — adding models' outputs together to produce a stronger

signal. The v2 architecture requires the counterintuitive insight that in this specific problem, the second model's most valuable function is not to agree with the first model but to challenge it. This is the architectural innovation that the evidence fusion chapter of the working paper documents as its central contribution.

Architectural Insight

Evidence fusion is most valuable not when models agree, but when they disagree. Agreement confirms. Disagreement discriminates. The correct fusion architecture for a surveillance-plus-trajectory system uses the trajectory model to adjudicate surveillance alarms — removing false alarms, not piling onto confirmed ones.

Chapter 11 Governance Architecture and Audit Infrastructure

The governance architecture of this project was not an afterthought. It was a design principle imposed from the beginning. Every model was required to produce auditable artifacts before its outputs were accepted as evidence. Every performance claim was required to be supported by a frozen file, a reproducibility check, or an execution trace. Every governance failure was documented and corrected rather than concealed.

11.1 The C6 Governance Framework

The OCC 5C quality dimensions framework was extended to a six-dimension C6 system for this project: Consistency, Coherence, Contradictions, Confabulation, Calibration, and Coverage. Every model was evaluated against all six dimensions before its results were accepted as verified. The C6 5/5 designation in the results tables indicates that a model achieved acceptable scores on all six dimensions.

The Dialogue Auditor component of the OCC tool performed a three-layer independent analysis of each AI system session: Layer 3 evaluated strategic and mission alignment (did the session outputs serve the stated mission?); Layer 2 evaluated narrative and claim provenance (was each numerical claim computed, uploaded, or asserted without source?); and Layer 1 applied the standard 5C surface analysis. Special FATAL triggers were reserved for cited files that were never present in the session and computation accounts that changed between turns when challenged.

11.2 Documented Governance Events

This section documents the governance events that emerged during the project. The framing here is not accusatory — both Claude and ChatGPT contributed genuine, substantial value throughout the study, and the human architect could not have completed this work at this speed or quality without them. The events documented below are presented because they are instructive: they reveal the specific boundaries where human oversight added irreplaceable value, and they demonstrate that the OCC 5C governance framework caught errors that would otherwise have propagated silently into the published results.

The collaboration model that emerged over the course of the project can be described as governed acceleration. Both AI systems took direction, wrote and debugged code, implemented complex statistical models, built dashboards, evaluated model performance, and helped the human architect navigate technically demanding validation problems. The entire project — dataset design, five models, fusion architecture, economic scoring harness, twenty-one governance discoveries, HTML dashboard, working paper, OCC tool upgrade, WBS, and risk register — was completed in less than one week of elapsed effort. That is not achievable without AI assistance. The governance architecture did not slow that process. It made the outputs trustworthy.

This shifts the project management paradigm materially. In a conventional AI development program, project management is primarily about schedule adherence — meeting milestones, tracking hours, managing scope. In this project, schedule was never the binding constraint. The binding constraint was quality: ensuring that every numerical claim was traceable, every model output was reproducible, every fusion logic was architecturally correct, and every governance event was documented and corrected. The OCC 5C tool became the central quality instrument. Running a Dialogue Audit at each session transition, maintaining the C6 certification standard across all five models, and using the CSV Auditor to verify results tables from source files — these activities defined the cadence of the project, not the Gantt chart.

The following governance events are documented not as failures of the AI systems but as demonstrations of the governance architecture working as designed. In each case, the human applied the OCC framework, identified the discrepancy, challenged the AI system, received a full acknowledgment, and corrected the architecture. The AI systems were partners in that correction process — not adversaries.

Governance Event 1. Code execution fabrication: In an early session, an AI system produced numerical results for a model without having executed the underlying code. When the human applied the OCC Dialogue Auditor and asked for the source execution file, the system cited a file

that had never existed in the session. The OCC audit returned FATAL on the claim provenance dimension. The corrected protocol required every numerical result to be accompanied by a materialized CSV output file before acceptance. This governance event directly motivated the intermediate-file materialization policy that became one of the six enforcement points of the RATIO C6 standard.

Governance Event 2. Circular authorship: In early builds, an AI system wrote both the modeling code and the audit trail that was intended to verify the modeling code independently. A system cannot audit its own outputs — the audit trail simply restated what the code claimed rather than verifying it against external reality. The corrected architecture required human review of all audit trail conclusions and independent execution traces before any performance claim was accepted. This event established the two-session rule: the Initiator and Reviewer in separate fresh AI sessions for auditor independence.

Governance Event 3. Population routing mismatch: An AI system applied the Stage123 behavioral routing taxonomy to models that had been estimated on the full mixed customer population rather than the routed subpopulations. The results were technically valid within their own logic but did not satisfy the mission specification, which required models to be estimated on the correct population segments. When the human raised this in a governance challenge session, the AI system acknowledged the limitation fully and cooperated in designing the corrected approach. The OCC Dialogue Audit returned WARN on the strategic drift dimension. This event demonstrated that AI systems will acknowledge architectural mismatches when challenged — but will not volunteer them proactively.

Governance Event 4. Economic dashboard double-counting: An AI-generated economic dashboard double-counted two action row types, inflating the false positive penalty approximately eight times — reporting a figure of $29 million rather than the correct $3.5 million. The error was invisible in the dashboard presentation but was identified when the human cross-checked the dashboard total against the independently computed scoring harness. This event established the CSV Auditor as a permanent tool: human verification of results tables must be performed against source CSV files, not against dashboard summaries that AI systems generate from their own outputs.

Governance Event 5. Fusion architecture logic inversion: The initial Sequential Fusion architecture used the trajectory model to escalate surveillance alarms rather than to filter them. The distinction — escalation versus filtration — is the entire difference between a fusion architecture that inherits its worst component's false positives and one that eliminates them. The human architect identified the logic inversion through analysis of the decision logic, raised it as a

governance challenge, and the AI system immediately understood the error and collaborated in designing the corrected v2 architecture. Net economic value improved by $1.3 million as a direct result. This event is the clearest demonstration in the study of the value of human architectural authority: the AI system implemented the specification correctly — but the specification itself required human architectural insight to correct.

Governance Event 6. Evidence accumulation creating artificial certainty: The initial Stata MSM implementation accumulated weekly WoE signals over time, allowing many weak signals to compound into high-confidence departure classifications without any single week providing meaningful evidence. The AI system's own threshold sensitivity output files provided the data that triggered human review — an instance of the machine watching the human's prior design choices. The human architect identified the architectural problem against Jeffrey's scale, specified the corrected maximum post-burnin WoE approach, and the AI system implemented it. This became Discovery 21 and improved MSM net economic value from −$1.5 million to +$656 thousand.

The pattern across all six events is consistent: the AI systems implemented specifications with high competence, acknowledged errors when challenged, and cooperated fully in corrections. The human architect provided architectural authority, challenge function, and evidence verification — the three irreducible functions described in Section 11.3. The OCC 5C tool provided the structured framework that made each governance challenge systematic rather than intuitive. The combination of AI capability, human governance, and quality tooling is what made the project possible at the speed and quality it achieved.

11.3 The Three Irreducible Human Functions

The governance events documented above collectively demonstrate that AI systems in this project required human oversight at three irreducible points.

Architectural authority: the human architect defined the behavioral taxonomy, the Fourier threshold, the IOR abstention policy, the LSTM design specification, and the fusion decision logic. No AI system was authorized to make architectural decisions. When architectural decisions were implicitly made by an AI system (the v1 fusion logic), they required human identification and correction.

Governance challenge: the questions 'where is the execution file?', 'were the models re-estimated on the correct subpopulation?', and 'does the fusion logic filter or only escalate?' were asked by the human architect, not generated by either AI system. Both systems acknowledged errors when challenged but did not disclose them proactively. Human challenge function is not optional.

Evidence verification: the determination that a cited file had never existed, that a dashboard had double-counted rows, and that a fusion architecture had an inverted logic required external verification against actual files and actual code. AI systems cannot verify their own outputs against reality. Only the human can confirm that the claimed file exists, the code executed, and the logic is correct.

Chapter 12 Twenty-One Governance Discoveries

The twenty governance discoveries documented in this chapter represent the accumulated institutional learning of the research program. Many emerged unexpectedly. Several challenged conventional machine-learning assumptions. Together they explain how the project evolved from prompt-only reasoning into a simulation-driven, evidence-fusion architecture grounded in governance, economics, and hybrid cognition.

Discovery 1 — Customer Departure Is an Intervention Problem, Not a Classification Problem

The project began with the implicit assumption that departure could be treated as a binary classification task. Over time this view became inadequate. Organizations do not merely wish to classify departures. They wish to prevent them. The problem is fundamentally an intervention optimization problem: given a signal of departure risk, what action, taken at what cost, at what point in time, maximizes expected economic recovery? This reframing influenced every subsequent design decision.

Discovery 2 — Prompt-Only Reasoning Cannot Satisfy Governance Requirements

Early experiments demonstrated that large language models could generate plausible explanations of customer deterioration. However, prompt-only approaches produced no reproducible, frozen, auditable outputs. The same question asked in different ways produced different answers. No performance claim could be externally verified. Prompt reasoning proved useful for hypothesis generation and architectural design but categorically insufficient as a standalone governance mechanism for operational AI. The lesson is not that AI reasoning is unreliable — it is that reasoning must be grounded in materialized, verifiable artifacts before it can serve as auditable evidence.

Discovery 3 — Customers Are Time Series, Not Rows

Many business analytics systems reduce customers to single-row summaries. This project demonstrated that meaningful departure signals emerge through behavioral trajectories across time, not from static snapshots. A customer's current order quantity is nearly uninformative without knowledge of its trajectory over the preceding weeks. This observation justified the shift to sequence modeling and motivated the 16-week sliding window feature architecture.

Discovery 4 — Observation-Level Scoring Penalizes Early Detection

When model performance is measured at the weekly observation level, a customer flagged as departing twelve weeks before actual departure appears to generate twelve consecutive false positives followed by one true positive. From a business perspective, that twelve-week early warning is extremely valuable. From an observation-level accuracy metric, it looks like poor performance. The project transitioned to customer-level evaluation with lead-time economics to correctly credit early detection.

Discovery 5 — Detection Timing Often Dominates Statistical Accuracy

A model that identifies a departure one week before confirmed departure creates minimal recovery value. A model that identifies the same departure sixteen weeks earlier creates substantial recovery value. The exponential decay in the recovery probability formula quantifies this: each additional week of lag reduces recovery probability by 2.5 percent. Models should be evaluated and selected primarily on their detection timing distributions, not merely on their confusion matrices.

Discovery 6 — False Positives Are Not Equal

Traditional machine learning treats all false positives identically. In this project, a false positive that triggers a FLAG action costs $1 thousand. A false positive that triggers an INTERVENE action costs $40 thousand. The same confusion matrix entry — FP — represents 40× different economic consequences depending on the action intensity. The False Positive Penalty metric, which weights false positives by intervention cost, replaced raw FP counts throughout the project.

Discovery 7 — Behavioral Heterogeneity Requires Routing Before Modeling

A single anomaly detector applied to a population containing continuous, periodic, and idiosyncratic customers will systematically misclassify large fractions of the population. Periodic customers have inherent zero-order weeks that look like anomalies to a continuous-customer

detector. Idiosyncratic customers have random gaps that look like signals to any pattern-matching model. The Stage123 routing architecture — classify behavior first, then apply the appropriate detector — was the single most effective structural change in the project.

Discovery 8 — Fourier Analysis Is Underutilized in Commercial AI

Many practitioners focus exclusively on time-domain features. The project demonstrated that frequency-domain analysis via Fast Fourier Transform provides uniquely valuable information for periodic-purchasing customers. Amplitude dropout in the dominant purchasing frequency — the signal that a customer's ordering cycle is weakening — preceded obvious quantity decline in several departure cases and provided detection signals unavailable to time-domain models.

Discovery 9 — Inter-Order Residuals Are Powerful for Irregular Customers

For idiosyncratic customers, conventional volume-based anomaly detection fails because the customer's baseline is itself irregular. The Inter-Order Residual (IOR) approach — monitoring the gap since the last order against the customer's own historical average — provided a simple, robust signal for customers who would otherwise confound every pattern-matching model. The IOR framework produced the abstention policy: when the ordering pattern is too irregular to distinguish signal from noise, the responsible action is a low-cost FYI flag, not an expensive autonomous intervention.

Discovery 10 — Evidence Accumulates; Single Signals Mislead

Individual anomalous weeks rarely provide reliable departure signals. A customer who has one unusually quiet week has not necessarily begun departing. The Bayesian Weight of Evidence accumulation framework — inspired by I.J. Good's (1950) work on sequential evidence integration — converted weak weekly signals into robust multi-week evidence assessments. The discovery that accumulated evidence substantially outperforms point-in-time anomaly detection reshaped the entire scoring architecture.

Discovery 11 — Behavioral States Can Be Latent

The Hidden Markov Model phase introduced the concept that customer departure risk is a latent state — not directly observable, but inferable from a pattern of observable signals over time. Although no single HMM configuration proved definitively superior, the latent-state framework provided a useful conceptual bridge between observation-level anomaly detection and the trajectory-learning approach of the LSTM challenger.

Discovery 12 — Anomaly Detection Is Not Departure Detection

Isolation Forest experiments revealed an important distinction. Many statistically anomalous customers never depart. Many departing customers pass through phases that appear relatively normal before departure becomes obvious. Anomaly and departure are correlated but distinct phenomena. This distinction motivated trajectory learning: rather than asking 'is this customer unusual?', the LSTM asks, 'does this customer's behavioral trajectory resemble known departure patterns?'

Discovery 13 — Domain Knowledge Can Substitute for Historical Labels

Perhaps the most practically important discovery: the organization could describe plausible departure mechanisms in enough detail to simulate them, even without extensive historical departure records. This is the simulation advantage. An organization with deep knowledge of its customer relationships and typical attrition patterns can leverage that knowledge to generate training data that no historical record could provide — because historical records of rare events are, by definition, rare.

Discovery 14 — Simulation Can Be a Primary Development Path

Conventional AI development responds to data limitations by acquiring more data. This project demonstrated an alternative: behavioral simulation can sometimes provide more value per development dollar than additional historical data collection. The critical conditions for simulation to succeed are accurate behavioral assumptions, realistic parameter calibration, and validation against genuinely hidden ground truth. When these conditions are met, simulation compresses the time to a deployable, governed AI system.

Discovery 15 — Synthetic Data Improves Transparency

Many practitioners assume that synthetic data weakens validation by introducing artificial assumptions. The opposite occurred in this project. Synthetic episode generation forced every assumption about departure behavior into an explicit, inspectable, discussable specification. The departure trajectory parameters — depth, slope, zero-week ramp rate — became auditable documentation of what the model was trained to recognize. This improved transparency, not reduced it.

Discovery 16 — Trajectory Recognition and Surveillance Are Complementary, Not Competing

Stage123 and the LSTM challenger solved different aspects of the departure detection problem. Stage123 identified unusual behavior relative to a customer's own baseline. The LSTM identified behavioral sequences that resemble known departure trajectories. These are different capabilities. An organization needs both: early warning from surveillance and confident action from trajectory confirmation. Treating them as competitors produced inferior results to treating them as complementary evidence sources.

Discovery 17 — High AUC Does Not Guarantee High Economic Value

The empirical result that the architecture with the highest AUC (0.858) produced the lowest net economic value (+$1.3 million) while the architecture with the highest AUC (LSTM+Fourier, 0.858) produced net economic value of only +$1.3 million, while the Sequential Fusion v2 — not a pure surveillance model — produced the highest net value of +$3.3 million is the most counterintuitive finding of the study. It is also the most actionable. Enterprise AI selection processes that rank models by AUC are systematically miscalibrated for business deployment. Economic evaluation frameworks are required.

Discovery 18 — Governance and Performance Are Complementary, Not Opposing

A persistent misconception in enterprise AI is that governance imposes costs on performance — that audit trails, reproducibility checks, and ground truth isolation slow development and reduce model capability. This project found the opposite at every stage. Governance constraints forced better architectural decisions: the requirement for frozen outputs discovered the fabrication problem; the requirement for reproducibility checks discovered the degenerate LSTM classifier; the requirement for explicit audit files discovered the double-counting error.

Discovery 19 — Fusion Is Most Valuable When Models Disagree

The corrected Sequential Fusion architecture is most valuable precisely where Stage123 and the LSTM disagree — where Stage123 flags a customer as departing but the LSTM cannot confirm a departure trajectory. Those disagreements are the false positives. A fusion architecture that responds to disagreement by maintaining the alarm inherits false positives. A fusion architecture that responds to disagreement by removing the alarm filters them.

Discovery 20 — Hybrid Cognition May Be the Most Durable Architecture

The final and most broadly applicable discovery is that effective decision support in complex environments may require the structured integration of multiple evidence streams, multiple reasoning architectures, and human judgment — rather than dependence on any single model, however sophisticated. Statistical reasoning, probabilistic surveillance, sequence learning, and human architectural authority each contribute forms of knowledge that the others cannot replicate. The hybrid cognition architecture that emerged from this project is more robust, more auditable, and more economically valuable than any individual component operating alone.

Discovery 21 — Evidence Accumulation Can Create Artificial Certainty

One of the most practically important governance discoveries emerged during review of the Stata Markov Switching Model implementation. The initial version accumulated weekly positive Weight of Evidence signals over time — a seemingly reasonable approach that mirrors the Bayesian principle of evidence integration. However, review against Jeffrey's scale revealed a structural problem: the accumulation mechanism allowed many weak signals to compound into a high-confidence departure classification, even when no individual signal exceeded the level of meaningful evidence. A customer generating a modest 2 dB WoE signal every week for twenty-five consecutive weeks would accumulate 50 dB — a level that Jeffrey's scale classifies as Decisive Evidence — without any single week providing meaningful departure evidence.

This is not a genuine departure signal. It is an artifact of the accumulation architecture. The corrected v2 implementation addresses this directly by using the maximum post-burnin leaving probability observed for the customer, converted to WoE and compared against a threshold policy ladder. This eliminates the compounding effect. A customer must achieve a meaningfully high departure probability in at least one week to trigger an action — not merely produce a long string of marginally elevated readings.

The governance implication extends beyond this specific model. Any evidence accumulation system that sums positive signals without a commensurate decay or floor mechanism is susceptible to this failure mode. The discovery reinforces the principle that governance of AI systems must include review of the mathematical architecture of the scoring system itself — not merely the inputs, outputs, and audit files. In this case, the machine's own threshold sensitivity output files provided the evidence that triggered human review of the accumulation logic — an instance of the machine watching the human's prior design choices and flagging an architectural error.

Chapter 13 Implications for Enterprise AI Investment Strategy

The findings of this study have direct implications for how organizations should think about enterprise AI investment, development methodology, and deployment governance. Three implications stand out as most practically significant.

13.1 The Data Cleanup Budget Requires Reexamination

The prevailing wisdom in enterprise AI is that data quality is the primary bottleneck. Organizations routinely allocate the majority of their AI development budgets to data cleaning, data integration, and labeling exercises before any model is trained. This allocation is not irrational — poor data quality does degrade model performance. But it may be misallocated for a specific and underappreciated class of problems: problems involving rare, consequential events where historical examples are scarce, expensive to label, or distributed across systems that resist integration.

For such problems, simulation-driven development offers a different investment logic. The core investment is in behavioral scenario design: expert workshops that articulate the mechanisms preceding the outcome of interest, parameterization exercises that calibrate synthetic episode characteristics to observable statistical properties of the real population, and validation exercises that test synthetic-trained models against hidden ground truth. This investment is qualitatively different from data cleanup. It draws on domain expertise rather than data engineering capacity, produces auditable documentation rather than clean tables, and scales with organizational knowledge rather than data volume.

The implication is not that data cleaning budgets should be eliminated. It is that organizations facing rare-event AI problems should explicitly evaluate the simulation alternative before committing to extensive data cleanup programs. The question to ask is: can our domain experts describe the mechanisms that precede the outcome we care about with enough precision to parameterize a synthetic episode generator? If the answer is yes, simulation-driven development may deliver a deployable, governed AI system faster and at lower cost than historical data assembly.

13.2 Architecture Selection Requires Economic Specification

The horse race results demonstrate that architecture selection cannot be separated from economic specification. The same six architectures produced net economic values ranging from

−$119 thousand to +$3.3 million under the locked formula on the same dataset. The ranking of architectures changed depending on whether EDP, FPP, or NEV was the objective. The corrected fusion architecture produced the best precision and near-optimal NEV but not the best EDP.

For enterprise deployment, this means that the architecture selection decision requires an explicit specification of the deployment economics before model evaluation begins. What is the average intervention cost at each action level? What is the expected revenue recovery under each intervention type? What is the expected lead time distribution for departure events in the target customer population? Without these inputs, model comparison produces meaningless rankings.

The RATIO AI Audit Practice recommends that every enterprise AI deployment begin with an economic model specification — not as a post-hoc evaluation exercise, but as the primary design constraint that determines which model architectures are worth pursuing. This reverses the conventional development sequence: specify the economic objective first, select candidate architectures consistent with that objective, and evaluate against economic criteria rather than statistical accuracy metrics.

13.3 Human Oversight Is an Architectural Element

The three irreducible human functions identified in Chapter 11 — architectural authority, governance challenge, and evidence verification — are not compliance requirements imposed by regulators. They are structural necessities that emerge from the capabilities and failure modes of current AI systems. AI systems can generate plausible explanations, implement specifications, produce statistical outputs, and adapt to feedback. They cannot currently define their own architectural constraints, challenge their own outputs when those outputs are wrong, or verify their assertions against an external reality they do not have access to.

The human-in-the-loop design of the governance architecture in this project was not a concession to regulatory pressure, nor did it slow the project down. The entire development program was completed in less than one week of elapsed effort precisely because AI systems handled implementation and the human architect handled architecture, challenge, and verification. Governance and speed are not in tension when governance is designed into the workflow rather than appended to it after the fact. It was the recognition that the human architect was the system's only reliable source of architectural authority, challenge function, and external verification. Removing the human from any of these three roles would have propagated errors — the fabrication event, the fusion logic inversion, the double-counting error — through to the final results without correction.

The implication for enterprise AI programs is that human oversight must be designed into the AI system architecture, not added as a procedural requirement after the system is built. The governance architecture must specify which decisions require human approval, what evidence the human needs to make those decisions, how the human's decisions are recorded in the audit trail, and how the system behaves when human review is unavailable. These are design questions, not compliance questions.

Chapter 14 Connections to PRIMMS-GPT and Project Management

The governance architecture developed in this project has direct connections to project management methodology, and specifically to the PRIMMS-GPT framework developed by Milestone Planning and Research, Inc. PRIMMS-GPT integrates phase-gate quality review, governing equation thresholds, and LLM-assisted project intelligence into a structured project management system. The customer departure detection project is itself a case study in the application of PRIMMS-GPT governance principles to an AI development program.

14.1 The Project as a Managed Program

The departure detection project followed a recognizable program structure: an initiation phase (problem definition, dataset design, mission specification); a development phase (model construction, iterative testing, governance event documentation); a validation phase (horse race execution, economic scoring, fusion architecture comparison); and a deployment preparation phase (HTML dashboard, working paper, audit package). Each phase had defined deliverables, quality gates, and handoff documentation.

The transition between Claude and ChatGPT sessions followed the PRIMMS-GPT handoff protocol: a structured handoff summary document was produced at each transition, specifying the current state of all verified results, the locked parameters, the next build task, and the governance record to be carried forward. This documentation structure prevented the loss of institutional memory across AI system transitions — a common failure mode in multi-session AI development programs.

14.2 Risk Identification from the Governance Discoveries

The twenty governance discoveries documented in Chapter 12 translate directly into a risk register for AI development projects. Each discovery identifies a failure mode that occurred in this

project and can be anticipated in future projects. The following risk categories emerged from the governance record:

- Data and label risks: ground truth leakage, observation-level evaluation bias, heterogeneous population contamination
- Model development risks: degenerate classifier from single-class training, AUC-NEV divergence, threshold calibration after ground truth exposure
- AI system behavior risks: fabrication, circular authorship, proactive non-disclosure of limitations, architecture logic inversion
- Economic evaluation risks: false positive double-counting, equal-cost false positive assumption, lead-time discounting omission
- Fusion architecture risks: escalation-only fusion inheriting surveillance FP count, confirmation threshold miscalibration

A formal risk register derived from these categories, formatted for PRIMMS-GPT ingestion, is available as a companion deliverable to this paper. Each risk entry includes a risk identifier, description, likelihood, impact, detection indicator, and recommended mitigation — structured as PRIMMS-GPT risk entries for direct integration into project monitoring workflows.

14.3 The Work Breakdown Structure

The project work breakdown structure, reconstructed from the actual development sequence — which spanned less than one week of elapsed effort for the full study — provides a template for future AI development projects following the simulation-driven governance architecture. The top-level WBS contains five phases: (1) Mission and Dataset Design, (2) Behavioral Taxonomy and Routing Architecture, (3) Platform Model Implementation, (4) Challenger and Fusion Development, and (5) Governance Audit and Publication. Each phase decomposes into specific work packages with defined inputs, outputs, and quality gates. The critical observation is that quality gates, not time milestones, governed phase transitions. A phase was complete when its CSV audit passed, its OCC review returned PROCEED, and its locked results were verified against the scoring harness — not when a scheduled date arrived.

A detailed WBS formatted for PRIMMS-GPT import, including task identifiers, work package descriptions, predecessor dependencies, estimated effort, and quality gate criteria, is available as a companion Excel deliverable to this paper.

14.4 Project Lifecycle: POC, Environment Setup, System Testing, UAT, and Controlled Deployment

The departure detection project has progressed through the first of five distinct lifecycle stages. Each stage has different success criteria, different human roles, and different prerequisites. Understanding the distinction between these stages is as important as understanding the models themselves — each stage answers a different question about the system, and no stage can be entered until the preceding stage is complete.

Stage 1 — Proof of Concept — is complete. The POC verified that the models work, the governance architecture is sound, the economics are verified, and every number traces to a source CSV. The 178-tab Customer Validation Workbook confirmed zero discrepancy across all 175 customers and all three model architectures within the research environment. The POC answers the question: do the models produce correct, auditable, reproducible outputs in the environment in which they were built? It does not answer whether the system is ready for operational use, nor whether it will behave correctly when migrated to a different platform and data environment.

Stage 2 — Environment Setup — is the current project status. Before any formal system testing or UAT can occur, the software and data must move to appropriate environments. The specific platform, licensing, and infrastructure decisions for each model component will be evaluated in this stage in consultation with IT and the relevant platform vendors. Representative data — either a sanitized subset of real customer records or a recalibrated synthetic dataset — must be staged in the target environment. IT approval of the platform stack is a prerequisite for Stage 3, not a parallel track. The OCC governance framework, audit file generation requirements, and human sign-off protocols must be configured for the new environment before any testing begins.

Stage 3 — System Testing in the Target Environment — follows successful environment setup. Once the stack is deployed on IT-approved infrastructure with representative data, formal system testing re-verifies that models produce correct outputs in the new environment. This is not a repeat of the POC validation workbook. It is a confirmation that migration introduced no errors, that the scoring harness operates correctly on representative data, and that audit file generation is intact. The specific activities for system testing are:

First, confirm that each model component produces identical decisions on representative data in the new environment as it produced on equivalent synthetic data in the POC environment. Any divergence is a migration error, not a model error, and must be resolved before proceeding. Second, verify that the scoring harness reads input files, applies the locked economic formula, and writes output CSV files correctly in the new runtime environment. Third, confirm that all audit

trail files — governance logs, threshold sensitivity outputs, reproducibility checks — are generated correctly and can be reviewed by the OCC framework in the new environment.

Stage 4 — User Acceptance Testing — follows successful system testing. UAT is operationally focused, not technically focused. It answers a different question: will this system meet users' needs in practice? The specific UAT activities for this system are: account managers and customer success professionals must evaluate whether the departure risk scores are actionable in their actual workflow; the action ladder parameters must be recalibrated against the organization's real intervention costs and recovery probabilities; and the weekly scoring cadence and reporting format must be evaluated against existing operational rhythms. A model that generates departure risk scores weekly but whose outputs are reviewed monthly will miss early-detection value. UAT requires real users in a representative operational context and cannot begin until Stage 3 system testing has passed.

UAT for AI systems differs from conventional software UAT in one crucial respect: the acceptance criteria are probabilistic, not binary. A model cannot be certified as always correct. What can be certified is that the governance framework provides sufficient transparency and traceability to support confident operational use — that errors will be detected, documented, and corrected when they occur. The OCC 5C framework and the CSV Auditor are the primary instruments for that certification.

Stage 5 — Controlled Deployment — follows successful UAT. Controlled deployment means a limited production rollout: typically one business unit, one customer segment, or one geographic market. Success metrics must be defined before deployment begins — not after. A rollback protocol must be in place. OCC governance reviews must continue at each production cycle, with the same rigor applied during development. The transition from UAT to controlled deployment is not a one-way door: if production data reveals that model behavior differs materially from behavior on the synthetic development dataset, the system returns to development for recalibration. That is not failure — it is the governance framework working as designed.

14.5 Proof of Concept Designation and Deployment Pathway

This study is designated a Proof of Concept (POC). That designation carries a specific meaning in this context. The models work: the governance architecture is sound, the economic scoring is verified, the results foot to zero on all three model platforms, and every number traces to a source CSV. The POC designation does not qualify the evidence quality. It qualifies the environment. The entire development program ran on a MATLAB desktop installation, a Stata do-file executed on a licensed workstation, and Python scripts in a local environment. That is a research-grade

stack, not a production-grade stack. The distinction matters for any organization considering deployment.

Moving from POC to production requires explicit decisions about environment, hardware, and licensing for each of the three model components. Those decisions are not resolved in this paper. They are the next stage of the deployment pathway, and they are documented here so that the transition is entered with clear eyes rather than discovered after deployment begins.

Stata Markov Switching Model. Stata is a licensed statistical platform. Licensing terms, deployment architectures, and server options will be evaluated in consultation with StataCorp and IT during the environment setup phase. The governing equations produced by the MSM are documented and auditable; the specific path for operationalizing them in a production environment is a follow-on decision.

MATLAB LSTM Challenger. MATLAB is a licensed technical computing platform. MathWorks offers multiple deployment options for trained models; the appropriate path for the LSTM — whether through MATLAB Production Server, model compilation, or another mechanism — will be evaluated in consultation with MathWorks and IT during the environment setup phase. The trained network architecture and weights are fully documented within the POC environment.

Python Components (GAP Fourier Branch and Sequential Fusion). The Python scripts (GAP Fourier branch and Sequential Fusion) are open-source and platform-agnostic. The specific runtime environment, scheduling, and integration architecture will be determined during the environment setup phase in consultation with IT. The scoring harness produces a structured CSV per scoring cycle; the fusion decision file is the natural integration point for any downstream system or workflow.

Environment Separation. The transition from POC to production requires three distinct environments: Development (the current research-grade stack), Testing (a UAT environment using representative data on an IT-approved platform stack), and Production (licensed servers, monitored pipelines, rollback protocol, and defined ownership of the model in operation). The OCC governance framework must travel with the system through all three environments. It is not a development-only tool. Governance reviews, audit file generation, and human sign-off requirements apply in UAT and production with the same rigor as in development. The governance framework is what makes the system trustworthy at each stage — removing it in production to simplify deployment would eliminate the primary source of operational confidence.

The Equation Extraction Principle. The key message for IT and deployment stakeholders is that this architecture was built for transparency. The governing equations, model parameters, and scoring logic are fully documented and auditable at every step. The POC environment produces structured output files at each stage of the pipeline — no black boxes, no opaque inference engines. That transparency is a governance asset and an IT risk management asset: it means deployment decisions can be made from a position of full information rather than vendor dependency. The specific platform and licensing paths for production will be evaluated in the environment setup phase.

14.6 The Quality Management Paradigm Shift

The most significant project management lesson from this study is not about the models. It is about where project effort goes when AI systems write the code.

In conventional software development, effort is distributed across requirements, design, coding, testing, and deployment. The coding phase is typically the largest single consumer of calendar time and skilled labor. In this project, coding was not the bottleneck. Claude and ChatGPT implemented five complete model architectures — including a deep learning LSTM challenger trained on synthetic behavioral episodes — in a fraction of the time a conventional development team would require. The entire project, from dataset design through published working paper, was completed in less than one week of elapsed effort.

What the AI systems cannot do is certify their own outputs. This is not a limitation that will be resolved by more capable models. It is a structural property of any system that cannot independently verify its outputs against external reality. The MSM's fabricated Python results, the dashboard's double-counted FPP, the fusion v1's inverted logic — none of these were errors the AI systems flagged proactively. Each was identified by a human applying a structured governance challenge.

This produces a fundamental reorientation of project management priorities. When AI writes the code, the schedule risk almost disappears. The quality risk does not. The binding constraint shifts from time-to-build to confidence-in-results. Project management in this regime is therefore primarily quality management: maintaining a rigorous evidence trail, running systematic governance reviews at each session boundary, demanding proof for every performance claim, and verifying that numbers foot before they are published.

The OCC 5C framework, the CSV Auditor, the Validation Workbook, and the locked results protocol are the instruments of this quality management regime. They do not slow the project. They make the outputs trustworthy enough to release. In a world where AI can build fast, the

organizations that will capture the most value are not those with the fastest AI — they are those with the most rigorous quality management of AI output. Schedule adherence is no longer the primary project management challenge. Evidence quality is.

Chapter 15 The Inductive Enterprise and the Future of Simulation-Driven AI

The broader intellectual framework within which this project operates is the Inductive Enterprise — an organizational design philosophy in which decisions are grounded in accumulated probabilistic evidence rather than deterministic rules, and in which AI systems serve as evidence generators and hypothesis testers rather than autonomous decision authorities. The departure detection project is a single case study within this framework, but its lessons extend to any enterprise domain where outcomes are rare, consequences are material, and behavioral mechanisms can be described by subject matter experts.

15.1 The Inductive Enterprise Defined

The Inductive Enterprise, as developed in the broader RATIO research program, holds that complex organizational decisions require the structured integration of multiple evidence streams — statistical, behavioral, contextual, and judgmental — processed through explicit reasoning architectures that maintain audit trails and support human oversight. This is fundamentally different from the conventional approach of fitting a single model to historical data and accepting its outputs as authoritative.

The intellectual foundations of the Inductive Enterprise draw on several converging traditions. Bayesian inference, as developed by E.T. Jaynes and before him I.J. Good, provides the probabilistic framework for sequential evidence accumulation. Friston's cortical hierarchy model, applied beyond neuroscience to organizational cognition, provides a structural framework for understanding how higher-order representations constrain and contextualize lower-order evidence. Kanerva's sparse distributed memory provides an architectural metaphor for how pattern recognition systems can generalize from limited examples to novel instances. Together these frameworks suggest that effective enterprise intelligence requires not more data but better evidence architecture.

15.2 Simulation as Organizational Knowledge Capture

The simulation-driven development path demonstrated in this project represents a specific operationalization of the Inductive Enterprise framework. When an organization's domain experts can describe departure mechanisms precisely enough to parameterize synthetic episode generators, the simulation process is, in effect, converting tacit organizational knowledge into explicit, executable, testable form.

This conversion has governance value independent of its modeling value. The process of designing synthetic episodes forces experts to articulate and defend specific claims about departure behavior: what does a complete departure look like in weeks 1 through 16 after onset? What is the typical decline rate in market share erosion? How quickly do zero-week streaks accumulate in a slow fade? These questions may never have been asked in the organization before. Their answers, once captured in the synthetic episode specification, become an auditable, revisable organizational asset.

The implication for enterprise AI programs is that behavioral scenario design workshops — structured exercises in which domain experts articulate and parameterize plausible failure or departure mechanisms — may be as valuable as data engineering exercises. The output is not a clean dataset but a synthetic episode library: an explicit, testable specification of the organizational knowledge that would otherwise exist only in the heads of experienced account managers.

15.3 The NIST AI Risk Management Framework Connection

The governance architecture of this project aligns naturally with the NIST AI Risk Management Framework (NIST AI 100-1, 2023), which organizes AI risk management around four functions: GOVERN, MAP, MEASURE, and MANAGE. The governance events documented in Chapter 11 correspond directly to the MAP function — identifying and documenting AI risks before they cause harm. The C6 quality dimensions correspond to the MEASURE function — quantifying model performance across multiple dimensions. The economic scoring framework and human oversight architecture correspond to the MANAGE function — implementing controls that reduce risk to acceptable levels.

The GOVERN function — establishing organizational policies, accountability structures, and incentives for responsible AI — is represented in this project by the segregation of duties architecture (Claude vs. ChatGPT), the human architectural authority principle, and the documentation discipline that produced frozen output files and audit trails at every stage.

Organizations seeking to implement the NIST AI RMF in practice may find this project's governance architecture a useful reference implementation.

15.4 Future Research Directions

Several extensions of the simulation-driven governance architecture merit further investigation. The first is the application of the behavioral taxonomy and synthetic episode generation framework to other rare-event enterprise problems: equipment failure prediction, fraud detection, cybersecurity intrusion detection, and clinical deterioration monitoring all share the same structural characteristics — rare outcomes, behavioral precursors that can be described by experts, and costly interventions that must be precisely targeted.

The second is the development of more sophisticated fusion architectures. The corrected Sequential Fusion v2 architecture is a two-stage sequential design. More complex fusion architectures — Bayesian network integration of multiple evidence streams, reinforcement learning-based action selection under uncertainty, dynamic threshold adaptation based on feedback from resolved cases — may produce further improvements in both economic value and governance transparency.

The third is the integration of the simulation-driven development methodology into enterprise AI governance frameworks and standards. The NIST AI RMF, the EU AI Act, and emerging ISO standards for AI management systems all emphasize the importance of documentation, testing, and human oversight. The simulation-driven approach produces governance artifacts — synthetic episode specifications, training audit files, reproducibility checks, economic scoring documentation — that directly satisfy these requirements. A formal mapping between the simulation-driven methodology and emerging regulatory frameworks would facilitate enterprise adoption.

Chapter 16 Conclusions

This project began as a request to compare four AI modeling platforms against a customer departure detection problem. It ended as a demonstration of simulation-driven AI governance — a development methodology in which synthetic behavioral episodes, explicit economic scoring, and structured human oversight replace the conventional dependency on large volumes of historical labeled data.

The central empirical finding is clear: under the locked, traceable economic formula, four of the five evaluated architectures produced positive net economic value, with the fifth (Hidden Markov Model) producing marginally negative results. They produced it in fundamentally different ways, reflecting different operational philosophies — broad surveillance versus precise trajectory confirmation versus evidence fusion. No single architecture dominated on all dimensions. The correct architecture for a specific deployment depends on the economics of that deployment: the cost of intervention, the value of early detection, the operational capacity for follow-through.

The central methodological finding is equally clear: an organization that can describe the mechanisms preceding the outcomes it cares about can train a supervised AI model without waiting for a large historical sample of labeled examples. The knowledge required to build a competent departure detector was already present in the organization that designed this study. The simulation process converted that knowledge into executable training data. The governance process converted those training data into a blind challenger that achieved an AUC of 0.858 and near-zero false positive cost — without ever seeing a real departure label during training.

The twenty governance discoveries documented in Chapter 12 are, collectively, the most durable contribution of this work. Individual model results will become dated as new architectures emerge. The governance discoveries will remain relevant because they describe failure modes and corrective principles that are intrinsic to the structure of AI development with imperfect information, heterogeneous populations, and rare outcomes. Organizations encountering these problems — whether in customer retention, fraud detection, equipment maintenance, or clinical monitoring — will encounter the same failure modes and require the same corrective principles.

Three conclusions deserve particular emphasis for practitioners. First, the data cleanup investment must be evaluated against the simulation alternative for rare-event problems. Second, model selection must be grounded in economic specification, not statistical accuracy metrics. Third, human oversight must be designed into the AI system architecture as a structural element, not added as a compliance layer. These are not abstract recommendations. They emerged from specific, documented events in a real AI development program and are supported by specific, quantified economic consequences.

A fourth conclusion deserves emphasis for organizations considering where to begin with AI: anomaly detection and behavioral surveillance are the lowest-risk entry point. The requirements are modest — a clean transactional history and domain knowledge of what failure looks like. The outputs are interpretable and economically scorable. The governance is tractable. And the

investment is weeks, not years. Organizations do not need to build massive data cleanup programs or large bureaucratic governance structures before their first AI deployment. They need a clean signal, a domain expert who can describe the behavioral mechanisms they care about, and a quality framework that ensures they can verify the output before acting on it.

The architecture that emerged from this project — behavioral taxonomy routing, Bayesian evidence accumulation, trajectory learning on synthetic episodes, evidence fusion with human oversight, and AI-assisted development governed by the OCC quality framework — is not the final answer to the departure detection problem. It is a governance-first starting point: an architecture that is auditable, reproducible, economically justified, and designed for iterative improvement as new behavioral knowledge accumulates and new departure patterns emerge. That starting point is, in the end, what the enterprise AI field most urgently needs.

16.1 Limitations and Conditions for Generalizability

The findings of this study are subject to several limitations that future work should address. First, all results are based on a single synthetic dataset with one set of parameters (seed=42, 175 customers, 18 departures, 70/30 revenue concentration). Generalizability to other customer populations, industries, churn rates, and revenue distributions has not been empirically established. The simulation approach may perform differently in populations where departure mechanisms are less well-understood or where multiple departure types overlap substantially.

Second, the simulation approach succeeds in this study because domain experts could describe departure trajectories with sufficient precision to parameterize a synthetic episode generator. This condition will not hold in all settings. If expert descriptions of failure or departure mechanisms are incomplete, contradictory, or structurally wrong, synthetic training data will encode those errors. The model will then be well-calibrated to the wrong behavioral assumptions. Validation against hidden ground truth — as performed in this study — is essential to detect this failure mode before deployment.

Third, the economic parameters — the 0.65 base recovery probability, the 0.025 weekly decay rate, and the action ladder cost structure — were calibrated to the specific context of this study. Organizations deploying this framework must recalibrate these parameters against their own intervention histories and cost structures. Applying the framework with unvalidated parameters

will produce economically meaningless rankings. The architecture is general; the parameters are not.

References

Aaron, J. (2026). The Inductive Enterprise: Governing AI Through Evidence Architecture. Milestone Planning and Research, Inc. Working Paper Series.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Annals of Mathematical Statistics, 41(1), 164–171.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

Dempster, A. P. (1968). A generalization of Bayesian inference. Journal of the Royal Statistical Society, Series B, 30(2), 205–247.

Friston, K. (2010). The free-energy principle: A unified brain theory? Nature Reviews Neuroscience, 11(2), 127–138.

Good, I. J. (1950). Probability and the Weighing of Evidence. Charles Griffin and Co.

Good, I. J. (1985). Weight of evidence: A brief survey. Bayesian Statistics, 2, 249–270.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica, 57(2), 357–384.

Hansen, M., Perotti, V., Pérez-Suay, A., Camps-Valls, G., and others (2023). Reimagining synthetic tabular data generation through data-centric AI. arXiv:2310.16205.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Jaynes, E. T. (2003). Probability Theory: The Logic of Science. Cambridge University Press.

Kanerva, P. (1988). Sparse Distributed Memory. MIT Press.

Kim, C.-J. and Nelson, C. R. (1999). State-Space Models with Regime Switching. MIT Press.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. Proceedings of the IEEE International Conference on Data Mining, 413–422.

Liu, X. and David, I. (2026). Developing AI agents with simulated data: Why, what, and how? arXiv:2602.15816.

Lu, Y., Shen, M., Wang, H., and others (2023). Machine learning for synthetic data generation: A review. arXiv:2302.04062.

MIT News (2022). In machine learning, synthetic data can offer real performance improvements. MIT News Office.

National Institute of Standards and Technology (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1.

National Institute of Standards and Technology (2025). Digital Twins: A Framework for Enterprise AI Systems. NIST Special Publication Series.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257–286.

Shafer, G. (1976). A Mathematical Theory of Evidence. Princeton University Press.

Sutton, R. S. and Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 13(2), 260–269.

Bain and Company (1996). Customer Loyalty and the Economics of Retention. As cited in Reichheld, F.F. (1996). The Loyalty Effect. Harvard Business School Press.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.

Coussement, K. and Van den Poel, D. (2008). Integrating the voice of customers through call center emails into a decision support system for churn prediction. Information and Management, 45(3), 164–174.

Hadden, J., Tiwari, A., Roy, R., and Ruta, D. (2007). Computer assisted customer churn management: State-of-the-art and future trends. Computers and Operations Research, 34(10), 2902–2917.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284.

Kumar, V. and Reinartz, W. (2018). Customer Relationship Management: Concept, Strategy, and Tools (3rd ed.). Springer Texts in Business and Economics. ISBN 978-3-662-55380-0.

Neslin, S. A., Gupta, S., Kamakura, W., Lu, J., and Mason, C. H. (2006). Defection detection: Measuring and understanding the predictive accuracy of customer churn models. Journal of Marketing Research, 43(2), 204–211.

Netzer, O., Lattin, J. M., and Srinivasan, V. (2008). A hidden Markov model of customer relationship dynamics. Marketing Science, 27(2), 185–204.

Ng, A. (2021). A chat with Andrew on MLOps: From Model-centric to Data-centric AI. DeepLearning.AI. Presented at NeurIPS 2021 Data-Centric AI Workshop. Retrieved from https://www.datacentricai.org/neurips21/

Reichheld, F. F. and Sasser, W. E. (1990). Zero defections: Quality comes to services. Harvard Business Review, 68(5), 105–111.

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., and Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. European Journal of Operational Research, 218(1), 211–229.

Verbraken, T., Verbeke, W., and Baesens, B. (2012). A novel profit maximizing metric for measuring classification performance of customer churn prediction models. IEEE Transactions on Knowledge and Data Engineering, 25(5), 961–973.

Wangperawong, A., Brun, C., Laudy, O., and Paulus, R. (2016). Churn analysis using deep convolutional neural networks and autoencoders. arXiv:1604.05377.

Zhao, J., Tang, J., and Shi, R. (2019). Customer churn prediction in telecommunications: A time series approach. PLOS ONE, 14(6), e0214379. https://doi.org/10.1371/journal.pone.0214379

Gartner (2021, cited 2025). The Cost of Poor Data Quality. Gartner Research. As cited by IBM (2025): poor data quality costs organizations an average of $12.9 million annually. Retrieved from https://www.ibm.com/think/topics/data-quality

CustomerGauge (2025). Net Revenue Retention Benchmarks: B2B SaaS and Distribution. CustomerGauge Research Series.

Deloitte (2025). State of AI in the Enterprise 2026. Deloitte AI Institute. Survey of 3,235 business and IT leaders across 24 countries, conducted August–September 2025. Primary finding: insufficient worker skills is the leading barrier to AI integration; data quality and integration constraints cited as leading execution challenges preventing scale.

ISG (2025). AI Use Cases Double Though Business Outcomes Lag Ambition. Information Services Group. BusinessWire, September 15, 2025.

ISG (2025). AI in Production: Enterprise Deployment Benchmarks 2025. Information Services Group Research Report.

Recurly (2025). State of Subscriptions: Churn Benchmarks by Industry Segment. Recurly Research.

Recurly (2025). State of Subscriptions: Churn Benchmarks by Industry Segment. Recurly Research.

Sutton, R. S. and Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.

Qlik/ETR (2025). Qlik 2025 Agentic AI Study: Budgets Surge, but Data Readiness Delays Scale. Commissioned from Enterprise Technology Research (ETR). Survey of 200+ enterprise technology decision-makers, August 2025. BusinessWire, October 16, 2025.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 13(2), 260–269.

Appendix A Development Chronology

Phase	Name	Key Deliverable	Governance Lesson
1	Prompt-only reasoning	Hypothesis narratives	LLM prompts describe but do not audit
2	Executable model initiation	First CSV decision files	Frozen outputs required for governance
3	Ground truth isolation	Locked scoring harness	Models must not see evaluation labels
4	Weekly-to-customer correction	Lead-time economic model	Early detection must be credited
5	Stage123 routing	Three-path taxonomy	Behavioral heterogeneity requires routing first
6	Stata MSM build	Verified Stata results	Institutional platform choice is a governance decision
7	MATLAB HMM build	Verified MATLAB results	Cross-platform reproducibility as governance evidence
8	Python Isolation Forest	Highest NEV single model	AUC and NEV can diverge substantially

Phase	Name	Key Deliverable	Governance Lesson
9	Synthetic LSTM v1	Degenerate classifier	Single-class training produces trivial solution
10	Synthetic LSTM v2-v3	AUC = 0.858	Harder negative examples essential for boundary learning
11	GAP Fourier branch	81-customer periodic detection	Frequency domain captures cycle dropout
12	Sequential Fusion v1	+$2.0M NEV	Escalation-only fusion inherits surveillance FPs
13	Sequential Fusion v2	+$3.3M NEV	Filter fusion reduces FPP 92%
14	Governance documentation	Audit package	Twenty governance discoveries become institutional asset

Table A.1 Project development chronology and governance lessons

Appendix B Mathematical Specifications

B.1 Economic Recovery Model

Recovery value for a true positive detection:

$$V = 0.65 \times \text{intensity} \times \exp(-0.025 \times \max(0, \text{lag})) \times \text{annual\_revenue}$$

Where lag = detection_week − 53 (weeks from start of scored period to detection), annual_revenue is drawn from the customer registry, and intensity ∈ {0.40, 0.70, 0.90, 1.00} for {FLAG, OUTREACH, ESCALATE, INTERVENE}.

B.2 Bayesian Weight of Evidence

Weekly WoE in decibans:

$$\text{WoE}(t) = 10 \times \log_{10} [\, P(\text{departure} \mid \text{signal}_t) \times (1 - \text{prior}) \,/\, ((1 - P(\text{departure} \mid \text{signal}_t)) \times \text{prior}) \,]$$

Prior probability of departure = 0.10 (10% of customer population in a given monitoring period). Cumulative WoE with decay:

$$\text{CumWoE}(t) = \max(0, \text{CumWoE}(t-1) \times \text{decay} + \text{WoE}(t)) \quad \text{if WoE}(t) \geq 0$$
$$\text{CumWoE}(t) = \max(0, \text{CumWoE}(t-1) \times \text{decay}) \quad \text{if WoE}(t) < 0$$

B.3 Fourier Signal-to-Noise Ratio

FFT SNR for GAP customer periodicity classification:

$$\text{SNR} = 10 \times \log_{10} [\, \max(|F(f)|^2) \,/\, \text{median}(|F(f)|^2) \,] \quad \text{for } f > 0$$

Where F(f) is the discrete Fourier transform of the detrended baseline quantity series. PERIODIC threshold: SNR ≥ 4.77 dB.

Appendix C Key Terms and Abbreviations

Term	Full Name	Definition
AUC	Area Under the ROC Curve	Probability that the model ranks a randomly chosen departing customer above a randomly chosen stable customer.
C6	Six-Dimension Quality System	Extension of OCC 5C: Consistency, Coherence, Contradictions, Confabulation, Calibration, Coverage.
EDP	Economic Detection Potential	Gross value recoverable from true positive detections, discounted by detection lag and intervention intensity.
FFT	Fast Fourier Transform	Frequency-domain decomposition used to detect periodic purchasing cycles in GAP customers.
FPP	False Positive Penalty	Total intervention cost incurred by model actions on stable (non-departing) customers.
HMM	Hidden Markov Model	Probabilistic sequence model that infers latent behavioral states from multiple observable signals.
IOR	Inter-Order Residual	Gap between last order and current week, evaluated relative to customer's historical inter-order interval distribution.
LSTM	Long Short-Term Memory	Recurrent neural network architecture for learning long-range dependencies in sequential data.
MSM	Markov Switching Model	Two-regime time series model that estimates state-switching probabilities from behavioral observations.
NEV	Net Economic Value	EDP minus FPP: the net financial value created by model deployment, after intervention costs.
OCC 5C	Output Completeness Check	Quality framework evaluating AI outputs on Consistency, Coherence, Contradictions, Confabulation, and Calibration.
RATIO	Risk & AI Testing / Inquiry Outcomes	AI audit and validation methodology developed by Milestone Planning and Research, Inc.
Stage123	Three-Stage Routing Architecture	Classify behavior (Stage 1) → assess periodicity (Stage 2) → apply appropriate detector (Stage 3).
WoE	Weight of Evidence	Bayesian evidence measure in decibans quantifying how much a signal updates the probability of departure.

Table C.1 Key terms and abbreviations

Appendix D Locked Results — Final Horse Race

All results in this appendix are verified against the locked ground truth file (weekly_ground_truth_v2.csv, seed=42) using the external scoring harness. Stage123 results

carry C6 5/5 governance confirmation. LSTM Challenger results carry RunA/RunB reproducibility confirmation (98.9% agreement, 173/175 customers).

Architecture	Platform	AUC	TP	TN	FP	FN	EDP ($k)	FPP ($k)	NEV ($k)	C6
Markov Switching Model	Stata 18	0.5260	15	44	113	3	3,405	2,748	−1,539	5/5
Hidden Markov Model	MATLAB R2025b	0.5722	15	25	132	3	3,571	3,690	−119	5/5
Isolation Forest	Python 3.12	0.4391	15	22	135	3	3,866	3,680	+186	5/5
LSTM + Fourier Hybrid	MATLAB + Python	0.8581	16	138	18	2	1,567	280	+1,288	5/5
Sequential Fusion v1	Python 3.12	0.7399	16	19	138	2	—	—	—	EDP unverified
Sequential Fusion v2	Python 3.12	—	16	140	16	2	3,601	278	+3,323	5/5

Table D.1 Final horse race results — all architectures, locked ground truth, 175 customers

Dataset: 175 synthetic B2B customers, 156 weekly observations, seed=42, 18 departing customers (10.3%), 70/30 revenue concentration (Tier A). Economic model: recovery_prob = 0.65 × intensity × exp(−0.025 × lag). Action ladder: FLAG $1k / OUTREACH $7.5k / ESCALATE $17.5k / INTERVENE $40k. C065 (IDIOSYNCRATIC): ABSTAIN, $0 opportunity cost. All Stage123 results verified C6 5/5.

Governance Statement

These results were produced by an external scoring harness (ratio_score_v2.py) that read frozen model output files and applied the locked economic model to the locked ground truth file. No model had access to the ground truth file during construction, training, or inference. The scoring harness was maintained by a separate AI system (Claude) from the MATLAB model implementation (ChatGPT). Intermediate audit files are available upon request.