First Cross-Domain Validation of the Time Engine

A preregistered evaluation on historical FDIC banking data.

Document type: Technical Validation Report Version: 1.0 Author: Time Engine Technologies Published: July 2026

TIME ENGINE TECHNOLOGIES LLC

First Cross-Domain Validation of the Time Engine A Preregistered Evaluation on Historical FDIC Bank Failures

Verdict PARTIAL | AUROC 0.891 (95% CI: 0.869–0.910) Domain US Community Commercial Banks, FDIC Call Reports, 2000–2012 Prediction Horizon 4 quarters (1 year) Execution Date July 5, 2026

Abstract

The Time Engine is a domain-independent temporal state framework. Its core hypothesis is that meaningful predictive information about system failure can be extracted from any sufficiently observable complex system — without domain-specific training, calibration, or optimization. This paper reports the first preregistered, real-data evaluation of that hypothesis on a domain the framework had never previously encountered: historical US community commercial bank failures.

The engine implemented no banking equations. It incorporated no CAMELS logic, no banking heuristics, and no banking-specific optimization of any kind. Observable financial variables were translated into four canonical temporal state variables through a pre-specified mapping, and the sealed, frozen engine was applied to that representation alone. The experiment was preregistered: the protocol, the evaluation criteria, and the engine were all cryptographically frozen before any data was scored.

The engine achieved an AUROC of 0.891 (95% CI: 0.869–0.910), indicating strong discrimination between bank-quarters that eventually failed and those that did not — substantially outperforming a naive single-variable baseline. It fell short of a purpose-built CAMELS logistic regression baseline, and its discrete classification was not calibrated for this domain. Per pre-registered criteria, the result is recorded as PARTIAL. This paper documents the experiment, the results, and their honest interpretation. The result is consistent with the Time Engine's core hypothesis. It is not proof of it.

1. Introduction

The Time Engine is built around a single hypothesis: that a domain-independent temporal state framework, applied through a domain-specific observation mapping, can extract meaningful predictive information from a complex system it has never previously encountered. This hypothesis, if it holds across multiple domains, has significant implications for how predictive systems are designed and deployed across industries.

Testing that hypothesis rigorously requires applying the framework to a domain with no prior exposure — where the engine has received no training, no calibration, no optimization, and no domain-specific tuning of any kind. This paper reports the first such test. The domain selected was historical US community commercial bank failures, evaluated against FDIC Call Report data spanning 2000–2012.

The Time Engine had never been exposed to banking in any form. It knew nothing about Tier 1 capital ratios, CAMELS supervisory ratings, loan loss reserve methodology, or the regulatory framework governing bank examination. No banking equations were implemented. No banking heuristics guided the computation. Observable financial data was translated into a standard representation, and the sealed framework was applied to that representation exactly as it would be applied to any other system.

Why This Validation Matters

Most predictive systems are built by teaching a model everything about one domain. They are trained on historical examples, calibrated against domain-specific baselines, and optimized until they perform well on the problem they were designed for.

The Time Engine was designed differently. It attempts to recognize temporal state itself — the structural signature of how a system is evolving over time — rather than accumulating domain knowledge.

This validation asked a direct question: Can a temporal framework recognize impending failure inside a complex system it has never previously encountered?

This experiment provides initial empirical evidence that the answer may be yes.

The experiment was conducted under a preregistered protocol, with all design decisions, evaluation criteria, and the engine itself cryptographically frozen before any data was scored. The result — AUROC 0.891, PARTIAL verdict — is reported exactly as the pre- specified criteria define it.

The story of this experiment is not that the Time Engine predicted bank failures. The story is that a frozen, domain-independent temporal framework extracted meaningful predictive signal from a complex system it had never been taught to recognize.

This paper describes the experimental design, the results, and their significance — including where the framework succeeded, where it fell short of pre-registered success criteria, and what the result implies about the broader hypothesis under investigation.

---

2. Background: The Time Engine Framework

2.1 Architecture Overview

The Time Engine accepts four canonical temporal state variables. The engine's computation operates on this representation to produce a continuous temporal state score and a discrete band classification reflecting where the system sits in its temporal trajectory.

The engine's internal computation is sealed. Its mathematics, transformation logic, and signal processing are not exposed externally. What the framework exposes is an interface: standardized inputs in, temporal state assessments out.

2.2 The Domain Mapping Layer

The canonical mapping layer is the only domain-specific component in the architecture. For a given domain, an analyst specifies how observable domain variables translate into the four canonical temporal state variables. Once that translation is applied, the engine receives an abstract numerical representation and processes it identically regardless of the originating system type.

This separation is architecturally intentional. It means the engine's computation is not calibrated to banking, manufacturing, healthcare, or any other domain. The domain knowledge lives entirely in the mapping specification — not in the engine. The engine does not know what kind of system it is evaluating.

2.3 What This Validation Tests

This validation tests whether the engine, operating on a pre-specified canonical mapping of financial variables, produces a meaningful predictive signal for bank failure on data it has never seen. It does not test whether the framework is the best tool for this purpose. It tests whether a frozen, domain-independent temporal state model produces useful output when applied to a domain for which it was never designed.

3. The Domain-Generalization Hypothesis

The Time Engine is not a banking model that generalized to another domain. It is a domain-independent temporal framework whose first validation happened to be performed on banking.

The foundational question motivating this research is whether temporal degradation shares common structure across fundamentally different classes of complex systems. A turbine engine losing operational capacity over thousands of cycles, a financial institution accumulating deteriorating assets across fiscal quarters, a patient's organ function declining across hours of hospitalization — these systems differ in mechanism, measurement, and physical substrate. Do they share a detectable temporal signature?

The Time Engine framework is built on the hypothesis that they do. It proposes that the structural pattern of how systems approach failure — expressed through observable state variables — carries predictive information that is not domain-specific, and that a single sealed computation applied to a domain-appropriate canonical mapping can detect this pattern across system types.

This hypothesis is nontrivial. If it holds, it implies that domain-specific expertise, while valuable, is not the only source of predictive information about system failure. It suggests that temporal structure itself — the dynamics of how observable state evolves over time — contains a signal that generalizes across domains.

This experiment provides initial evidence relevant to that hypothesis. It does not prove it. Banking was selected as the first test domain specifically because of its contrast with mechanical systems: the failure mechanisms, measurement scales, regulatory context, and temporal dynamics of financial institutions are as different from turbine engines as a real-world domain can practically be. A positive result in this domain is more informative than a positive result in a structurally similar domain would be.

If the Time Engine's approach works on banking — a domain it was never taught — it becomes natural to ask whether it would work on engines, patients, supply chains, power grids, or any other complex system where state evolves observably over time. That question is the research program. Banking is the first data point.

---

4. Why This Result Was Unexpected

The conventional design philosophy for predictive systems is additive: performance improves as more domain knowledge is incorporated. Models are trained on historical examples from the target domain. Features are engineered by domain experts. Thresholds are calibrated against domain-specific baselines. The system improves because it learns more about the specific problem it is designed to solve.

The Time Engine deliberately inverts this logic. It incorporates no domain knowledge about banking. It received no training on historical bank failures. It was given no banking equations, no CAMELS logic, no banking heuristics, no domain-specific features, and no banking-specific optimization of any kind. The canonical mapping translated observable financial data into a standardized representation, and the sealed engine processed that representation exactly as it would for any other system.

Conventional predictive systems improve because they are given more domain knowledge. The Time Engine deliberately received less. It knew nothing about banking. Yet it still generated meaningful predictive discrimination.

The expected result for this design choice — applying a deliberately domain-agnostic framework to a domain it has never encountered, with no domain-specific optimization — is not AUROC 0.891. Domain-naive models applied to complex prediction problems typically perform near chance level or only marginally above simple baselines.

What is surprising is not the magnitude of AUROC 0.891 in isolation. What is surprising is the combination: a sealed computation, receiving no banking knowledge, producing strong out-of-sample discrimination on a domain it had never previously encountered. The conventional explanation for predictive power — that the model learned the domain — is not available here. The engine had nothing to learn from.

This does not mean the result proves the domain-generalization hypothesis. A single experiment in a single domain is initial evidence, not proof. But the result does falsify the simpler hypothesis that the framework would produce chance-level performance when applied to a new domain without domain-specific adaptation. AUROC 0.891, with a lower confidence bound of 0.869, is far above chance.

The Key Observation

The observed predictive signal cannot be attributed solely to trivial single-variable relationships. The engine substantially outperformed a naive single-variable baseline by 0.438 AUROC points. Something in the canonical temporal representation carries information about bank failure that a simple capital ratio threshold does not capture — even though the engine was never taught what bank failure looks like.

---

5. Experimental Design

5.1 Dataset

We evaluated the Time Engine against FDIC Call Report data for US community commercial banks — quarterly regulatory filings containing standardized financial data covering capital adequacy, asset quality, earnings, liquidity, and operational structure. The dataset was sourced from the Chicago Federal Reserve MDRM extract, covering 56 quarters from 1999Q1 through 2012Q4, spanning a full credit expansion cycle, the 2007– 2009 financial crisis, and the subsequent resolution period.

Dataset Parameter	Value
Source	Chicago Federal Reserve — FDIC Call Report MDRM Extract
Total bank-quarters	466,585 (raw); 343,998 (community-commercial cohort)
Total institutions	8,775 unique banks in community-commercial cohort
Evaluation window	2000Q1 – 2012Q4
Bank failures included	574 FDIC seizure events
Prediction horizon	k = 4 quarters (1 year ahead)
Scored evaluation set	166,968 bank-quarters; 737 confirmed failures (0.44% base rate)

Table 1. Dataset characteristics.

5.2 Canonical Observation Mapping

Before any scoring began, a domain expert specified how observable FDIC financial variables translate into the four canonical temporal state variables accepted by the engine. This mapping specification was frozen as a cryptographically hashed artifact. The engine was not involved in constructing this mapping, and the mapping was not adjusted after scoring began.

Once the mapping was applied, the engine received an abstract representation of system state. It had no access to the original financial variables, no knowledge of what they represented, and no ability to distinguish a bank-quarter from any other kind of system observation. It processed the translated representation and returned temporal state scores.

5.3 Baselines

Two baselines were pre-registered:

- B0 — Naive Capital Threshold: A binary flag based on a single capital adequacy ratio.

No machine learning. This baseline represents the simplest possible domain-informed

heuristic — one financial ratio thresholded. The engine must outperform it to establish that the canonical temporal representation adds value beyond a single-variable test.

- B1 — CAMELS Logistic Regression: A logistic regression model trained on the

CAMELS supervisory framework. Fit on a disjoint benchmark population; evaluated on the scored population. This represents the prevailing standard for this prediction problem. Non-inferiority against it was the most

stringent pre-registered success criterion.

5.4 Pre-Registered Evaluation Criteria

Outcome	Conditions Required	Interpretation
Full Success	AUROC ≥ 0.75 AND non-inferiority vs. B1 (ΔAUROC ≥ −0.02) AND monotonic calibration AND median lead ≥ 2 quarters	Framework performs competitively with the domain standard.
Kill Criterion	AUROC ≤ 0.60 OR beaten by the naive capital threshold (B0)	Framework produces no meaningful signal. Negative result.
Partial	Kill criterion not triggered. Full Success not achieved.	Framework shows genuine signal; does not yet match the domain standard.

Table 2. Pre-registered evaluation outcome definitions.

---

6. Protocol Integrity and Preregistration

6.1 Why Preregistration Matters

The validity of an empirical claim depends on whether the conclusion was predicted in advance or derived by exploring data after seeing the results. Predictive systems can be made to produce impressive metrics through iterative exploration — adjusting features, modifying evaluation windows, redefining success criteria after observing outcomes. Without a preregistered protocol, there is no way to distinguish a genuine a-priori prediction from a post-hoc rationalization.

We preregistered this experiment to produce an interpretable result. The AUROC of 0.891 is the result of the pre-specified experiment — not a selected result from a space of possible experiments explored after seeing the data.

6.2 What Was Frozen Before Scoring Began

Artifact	Description	Status
Validation Protocol	Cohort definition, evaluation criteria, baseline specifications, bootstrap parameters, and reporting standards.	FROZEN
Canonical Mapping Spec	The complete specification of how FDIC financial variables translate into canonical temporal state variables.	FROZEN
MDRM Translation Spec	Field naming convention translation for the Call Report data format.	FROZEN
Engine Specification	The sealed Time Engine computation, version-locked. Not inspected or modified during or after scoring.	FROZEN
Dataset Snapshot	FDIC Call Report MDRM extract, 466,585 bank-quarter rows.	FROZEN
Failure Outcomes File	574 FDIC seizure events with exact dates.	FROZEN
Protocol Amendments	Cohort filter and censoring specification — both ratified before scoring began.	FROZEN

Table 3. Artifacts frozen before scoring began.

What "Sealed Engine" Means

The Time Engine is a sealed computational engine. Its internal mathematics and transformation logic are not visible to the experimenters during execution. Scoring runs as a single end-to-end operation: canonical inputs in, temporal state scores out. There is no intermediate inspection step that could enable selective calibration or post-hoc adjustment.

6.3 Data Engineering Corrections

Two technical corrections were applied during execution, before final scoring, at the data extraction and loading layer only. Neither correction touched any frozen artifact. Both are disclosed in full.

Correction 1 — Entity Field Resolution: A field identifier for the institution certificate number was mislabeled in the dataset acquisition guide. The correct field was identified by cross-referencing the Federal Reserve institution crosswalk. Effect on results: zero. All computed metrics are identical with or without this correction, because it applies only to entity identification — not to any financial metric, canonical variable mapping, or score.

Correction 2 — CSV Quoting: The CSV parser did not initially handle RFC-4180 quoted fields, causing 136 of 574 failure records to be silently dropped during loading. When corrected, all 574 failure events were loaded. Effect on results: AUPRC increased (more true positive events recognized). AUROC was minimally affected. This correction fixed a measurement error. It did not change any scoring logic or frozen specification.

---

7. Results

7.1 Understanding AUROC

What AUROC Measures

AUROC — the Area Under the Receiver Operating Characteristic Curve — is a rank-based discrimination metric. Its value ranges from 0.50 (no better than random) to 1.00 (perfect discrimination between outcomes).

An AUROC of 0.891 indicates strong discrimination between bank-quarters that eventually failed and those that did not. The confidence interval of 0.869–0.910 means that even under conservative estimation (accounting for within-bank correlation across quarters), the lower bound remains well above both the kill criterion threshold (0.60) and the naive baseline (0.453).

Crucially: this discrimination was produced by an engine that implemented no banking- specific logic of any kind. AUROC is a rank-invariant metric — it measures the quality of the score's ordering, regardless of absolute calibration. The engine's rank ordering of bank- quarters by failure risk was strong, despite the engine's complete absence of banking knowledge.

7.2 Primary Results

Metric	Time Engine	B0: Naive Capital	B1: CAMELS Logistic
AUROC	0.891	0.453	0.964
95% Confidence Interval	0.869 – 0.910	–	–
AUROC Advantage over B0	+0.438	–	–
ΔAUROC vs. B1	−0.073	–	–
ΔAUROC CI vs. B1	−0.089 to −0.054	–	–
AUPRC	0.043	–	–
AUPRC Lift over Base Rate	~9.8×	–	–

Table 4. Primary discrimination metrics. CI computed via cluster-bootstrap (B=500), bank as resampling unit,

seed=20260705.

The Time Engine achieved an AUROC of 0.891, indicating strong discrimination between bank-quarters that eventually failed and those that did not. This was produced by an engine that had no prior exposure to banking — no banking training, no banking optimization, no banking heuristics — operating on a canonical representation derived from a pre-specified mapping of observable financial variables.

7.3 Kill Criterion Evaluation

- AUROC ≤ 0.60: Not triggered. The engine achieved 0.891 — 0.291 above this threshold.

The lower confidence bound (0.869) is itself far above it.

- Engine beaten by naive capital threshold (B0): Not triggered. The engine

outperformed B0 by 0.438 AUROC points (0.891 vs. 0.453). The canonical temporal representation substantially outperforms a single financial ratio.

This is not a negative result. The engine produces genuine, statistically robust out-of-sample discriminative power on a domain for which it received no preparation whatsoever.

7.4 CAMELS Comparison and Non-Inferiority

The CAMELS logistic regression achieved AUROC 0.964. The observed gap between the engine and B1 was −0.073 (CI −0.089 to −0.054) — entirely below the pre-registered non- inferiority margin of −0.02. CAMELS substantially outperforms the Time Engine on this cohort and horizon.

This result is scientifically expected. The CAMELS baseline was fit on a training population from the same economic period — it received calibration feedback from banking data. The Time Engine received none. The question is not whether a purpose- built, domain-calibrated model outperforms an uncalibrated domain-independent framework. The question is whether the domain-independent framework produces meaningful signal at all. AUROC 0.891 answers that question.

7.5 Band Calibration

The engine's discrete temporal band classification was not calibrated correctly for the banking domain. Approximately 93% of bank-quarters received the same band classification, rendering the binary flag operationally uninformative.

This is a known and expected consequence of applying a domain-independent engine to a new domain without domain-specific calibration constants. Band calibration and rank- order discrimination are separable properties. The AUROC result is not affected by this calibration issue — it is computed from the continuous temporal state score, not from discrete band assignments. The two populated band classifications do order correctly by failure rate, which is directionally consistent with the temporal framework.

7.6 Official Verdict

PARTIAL. The engine achieves AUROC 0.891 — strong discrimination indicating genuine predictive signal — and clears the pre-registered kill criterion by a wide margin. It does not achieve Full Success: it remains materially inferior to the CAMELS logistic baseline, and its discrete band classification is not calibrated for this domain. Recorded as PARTIAL per pre-registered protocol.

---

8. Scientific Interpretation

8.1 What Was Demonstrated

A sealed, frozen, domain-independent temporal state model — given no banking knowledge beyond a pre-specified canonical observation mapping — achieved an AUROC of 0.891, indicating strong discrimination between bank-quarters that failed and those that survived, on a disjoint out-of-sample population with no post-hoc modification of any kind.

The engine implemented no banking equations, no banking heuristics, no CAMELS logic, and underwent no banking-specific optimization or tuning at any stage. The canonical mapping translated observable financial data into a standardized representation; the engine processed that representation and returned temporal state scores. The predictive signal emerged from that sealed computation — from nothing else.

The observed predictive signal cannot be attributed solely to trivial single-variable relationships. The 0.438 AUROC point advantage over the naive capital baseline establishes that the canonical temporal representation contains information about bank failure beyond what a single financial ratio captures — even though the engine was never taught what bank failure looks like.

8.2 What Was Not Demonstrated

- That the Time Engine is the best tool for bank failure prediction. CAMELS logistic

regression outperforms it on this cohort and horizon.

- That the domain-generalization hypothesis is proven. One successful domain

generalization is initial evidence, not proof. Independent validation across multiple domains is required.

- That performance on this cohort and horizon generalizes to all banking conditions. This

evaluation uses a specific community-commercial cohort, a one-year prediction horizon, and a single historical period.

8.3 Why the Partial Result Increases Credibility

A preregistered experiment that reports a PARTIAL result is more scientifically credible than an unregistered experiment that claims complete success. The success criteria were defined before scoring. They required the engine to be non-inferior to CAMELS. It was not. The result is therefore recorded as the protocol specifies: PARTIAL.

If the success criteria had been defined after seeing the AUROC of 0.891, they would likely have been written to declare success. They were not. The value of the preregistration is precisely that it constrains what can be claimed. Future validations that achieve Full Success under similarly rigorous protocols will be more credible because this PARTIAL result was not retroactively reframed.

8.4 The CAMELS Gap: Interpretation

The gap between the engine and the CAMELS baseline reflects the deliberate design choice to exclude domain knowledge from the engine. The CAMELS model was calibrated on banking data; the Time Engine was not. Whether this gap represents a fundamental limit of the domain-generalization approach, or an implementation-specific gap addressable through domain-specific refinements, is an open empirical question. Future validation will provide data relevant to that question.

---

9. Why This Experiment Matters Beyond Banking

The significance of this result is not contained in its banking application. The significance lies in what the result implies about the framework.

Conventional predictive models are constructed around domain-specific expertise: credit risk models encode banking regulatory frameworks; clinical deterioration models encode medical scoring systems; predictive maintenance models encode engineering reliability concepts. Each model works because it accumulates knowledge about what failure looks like in its specific domain.

The Time Engine makes a different architectural bet: that failure — regardless of domain — has a recognizable signature in the temporal evolution of observable system state, and that this signature is accessible without domain expertise if the observation mapping correctly translates domain variables into a standardized representation.

The Time Engine is designed to operate on any sufficiently observable complex system. This validation provides initial empirical evidence supporting that hypothesis.

If this hypothesis holds across multiple domains, the implications extend well beyond banking. A single temporal state framework, applied through domain-specific observation mappings, could provide consistent temporal risk assessments across financial institutions, industrial equipment, patient populations, logistics networks, and any other class of system where state evolves observably over time — without requiring a purpose-built prediction model for each domain.

This experiment does not establish that outcome. It establishes that the first real-world test of the underlying hypothesis produced a result consistent with it. Banking happened to be the first proving ground. The natural question this result raises is not whether the Time Engine is a good banking model. The natural question is: what happens when you test it on engines? On patients? On supply chains? That question is the research program this paper opens.

---

10. Limitations

The following limitations are stated completely and without minimization.

- Material inferiority to the CAMELS baseline. ΔAUROC = −0.073, with a confidence

interval of −0.089 to −0.054 — entirely below the non-inferiority margin. CAMELS substantially outperforms the engine on this cohort and horizon.

- Band calibration failure. 93% of bank-quarters receive the same discrete band

classification. The binary band flag provides no useful discrimination. The continuous temporal state score discriminates well; the discrete classification system requires domain-specific calibration to function as intended.

- Single prediction horizon. Only k = 4 quarters was evaluated. Performance at shorter

or longer horizons is unknown.

- Single economic cycle. Failure events are concentrated in the 2008–2010 banking

crisis. Performance in banking stress scenarios of different character may differ.

- No domain-specific refinement evaluated. This evaluation deliberately used a domain-

independent engine configuration. The effect of domain-specific refinements has not been tested.

- Single domain. One cross-domain application is initial evidence, not proof of

generalization. Independent validation in other domains is required.

- Lead-time analysis not interpretable. The median first-flag lead time is a band

saturation artifact and does not represent genuine early detection capability in this evaluation.

---

11. Historical Significance

Before July 5, 2026, the Time Engine existed as a theoretical framework: a formal architecture, a set of patent claims, and design specifications. Whether the framework could produce meaningful predictions on real longitudinal data from a real complex system — one it had never previously encountered — had not been tested.

This validation is the moment that changed. A frozen, sealed computation — given no domain knowledge beyond a pre-specified canonical observation mapping — achieved an AUROC of 0.891 on historical bank failure data, with no post-hoc modification of any kind. No banking equations. No banking training. No banking optimization. The preregistration means this result cannot be attributed to exploration or selection. It is the result of the registered experiment.

This experiment represents the first transition from theoretical framework to experimentally evaluated system. Regardless of how future validations perform — whether they produce Full Success or additional PARTIAL results — this experiment establishes the empirical foundation on which further development is built. The hypothesis is no longer untested.

What This Experiment Establishes — Precisely

- The framework produces genuine out-of-sample signal. AUROC 0.891 with a lower

confidence bound of 0.869 is statistically robust. This is not noise.

- The canonical representation contains information beyond single variables. The

0.438 AUROC advantage over the naive capital baseline establishes that the temporal representation captures something beyond a simple threshold test.

- The sealed architecture produced real output. A sealed computation, receiving no

banking knowledge, produced strong rank-order discrimination. The preregistered protocol means this result is interpretable as a genuine a-priori prediction.

- Rigorous self-assessment is viable for a commercial research program. A

PARTIAL result was recorded as PARTIAL. This creates a verifiable scientific record that future work can build on.

---

12. Conclusion

We built a domain-independent temporal framework. We deliberately selected a domain we knew nothing about. We froze the engine. We preregistered the experiment. We refused to optimize after seeing the results. The framework nevertheless extracted meaningful predictive information from a complex system it had never previously encountered.

Banking happened to be the first proving ground. The Time Engine implemented no banking equations, no CAMELS logic, no banking heuristics, and underwent no banking- specific tuning or optimization of any kind. Observable financial data was translated into four canonical temporal state variables, and the sealed computation was applied to that representation alone.

The result was AUROC 0.891 — strong discrimination between bank-quarters that failed and those that survived — recorded as PARTIAL because it fell short of the pre- registered non-inferiority criterion against a calibrated CAMELS baseline. We drew three conclusions:

First: the canonical temporal representation contains predictive information about bank failure that is not reducible to a single-variable baseline. The 0.438 AUROC advantage over the naive capital threshold is robust and statistically significant.

Second: the domain-generalization hypothesis has not been falsified on this dataset. A frozen, domain-independent engine produced AUROC 0.891 on a financial domain it was never taught. That is initial evidence. It is not proof, and it should not be treated as such.

Third: scientific credibility requires honest reporting. The result was PARTIAL. We recorded it as PARTIAL. We did not reframe it, select a favorable subpopulation, or adjust the criteria after seeing the data. Future validations that achieve Full Success under similarly rigorous protocols will stand on a more credible foundation because of this.

Future work will evaluate additional domains.

This paper documents the first empirical validation — not the final word.

Appendix: Reproducibility and Protocol Integrity

This appendix summarizes the preregistered artifacts and dataset characteristics for independent verification. Researchers seeking to replicate these results should verify that they are working from the same frozen specifications using the identifiers below.

Artifact	Identifier / Hash
Validation Protocol	sha256: 41f630a8…a70c84 (registration and ratification commits on file)
Canonical Mapping Spec	metric-signal-map.fdic.v1.0.json | sha256: f2eeddc3…c579fd (canonical)
MDRM Translation Spec	fdic-field-normalization.mdrm.v1.json | sha256: d8920d6e…95556be (canonical)
Dataset Snapshot	mdrm_extract.csv | sha256: 73ef98a9…c664aa | 466,585 rows, 56 quarters
Engine Specification	spec.v0.1.0.json (sealed; internal hash not externally disclosed)
Bootstrap Configuration	B=500 cluster bootstrap, bank as resampling unit, seed=20260705
Protocol Amendments	Cohort filter (A1) and censoring specification (A3) — both frozen before scoring

Table A1. Frozen artifact identifiers.

Cohort Construction Summary

Filter Step	Records Remaining	Notes
Raw MDRM extract	466,585 bank-quarters	56 quarters, 1999Q1–2012Q4
Successful normalization	448,342 bank-quarters	18,243 dropped (missing required fields)
Community-commercial cohort	343,998 bank-quarters	8,775 unique banks
By-bank 50/50 split → scored pop.	4,401 banks	Disjoint from 4,374 benchmark banks; zero overlap
k=4 censored evaluation set	166,968 bank-quarters	737 positives; 0.44% base rate

Table A2. Cohort construction steps.

Notes on Replication

- The scored and benchmark populations are disjoint by bank. No bank appears in both.

This prevents temporal leakage and ensures the CAMELS baseline is evaluated on a truly out-of-sample population.

- The CAMELS baseline uses a point-in-time peer snapshot per evaluation date,

constructed from the benchmark population only. No data from the scored population enters the baseline fitting procedure.

- The two data engineering corrections described in Section 6.3 are loader-level changes.

Their nature and scope are disclosed in full. Replication should apply the same corrections.

- The engine specification is available under separate agreement with Time Engine

Technologies LLC for qualified research institutions.