

TIDE: A Statistical Process Control Framework for Governing Terminology-Induced Drift in Healthcare Machine Learning Operations

SHASHANK BHARADWAJ¹, AASTHA CHAUDHARY²

¹INDEPENDENT RESEARCHER, ASTA CRS INC, UNITED STATES

²INDEPENDENT RESEARCHER, AI ENGINEER, UNIVERSITY OF THE CUMBERLANDS, UNITED STATES

EMAIL: SHASHANKBHARADWAJ19@GMAIL.COM , CAASTHA03@GMAIL.COM

CORRESPONDING AUTHOR: SHASHANK BHARADWAJ

Abstract

Machine learning models deployed in healthcare read their inputs through administered clinical terminologies: ICD-10-CM diagnoses, CPT and HCPCS Level II procedures, LOINC laboratory observations, SNOMED CT concepts, and NDC drug identifiers. Most are revised by their maintainers on published calendars, although the cadences differ and NDC content turns over continuously with the drug market rather than on a fixed effective date. The fiscal year 2026 ICD-10-CM addenda, effective October 1, 2025, added 487 diagnosis codes, revised 38, and deleted 28. The April 1, 2026 update changed no codes at all, yet revised index entries that move documented neuroendocrine tumors from a benign category to a malignant one, altering severity capture without altering a single code. Release magnitude and release type therefore vary sharply between consecutive boundaries, and a release containing no code changes is not a null event for a deployed model. We define terminology-induced drift as distribution change whose proximate cause is a vocabulary revision rather than a change in clinical reality, and separate it into five mechanism classes. We then specify TIDE, which holds releases as versioned registry artifacts, monitors six terminology-sensitive statistics on control charts whose parameters stay fixed across release windows so that false alarm rates remain comparable, routes signals by whether they match a pre-registered release delta, and scores model exposure on a normalized index. A synthetic scenario on simulated Medicaid claims, run for 200 seeded replications, illustrates the mechanics: the unseen code rate chart signaled at a median of one day while the calibration CUSUM signaled within 120 days in only 4.5 percent of runs. These figures describe the generator, not field performance. A multi-site pilot protocol with an explicit minimum detectable effect is specified for empirical evaluation.

Keywords: machine learning operations, MLOps, dataset shift, concept drift, statistical process control, ICD-10-CM, clinical terminology, healthcare machine learning, model monitoring, HIPAA, change control, model risk management.

1. Introduction

Healthcare machine learning has a maintenance problem better documented than solved. Models that perform well at deployment degrade in production as the data around them changes, a phenomenon the literature calls dataset shift, and clinical environments produce it in abundance: new treatments alter outcome distributions, documentation practices evolve, patient populations move, and information systems are replaced [1], [2]. Finlayson and colleagues catalogued how these shifts erode deployed clinical models and argued that recognizing and correcting for shift is now a core competency for anyone fielding one [1]. What the field has not done is separate the drift that arrives unannounced from the drift that arrives on a maintainer publication schedule.

Consider what a model trained on coded data consumes, noting that the vocabularies differ in who maintains them and how predictably they change. Diagnoses arrive as ICD-10-CM codes maintained by the National Center for Health Statistics, with the annual update effective every October 1 and a secondary update every April 1 [18], [19]. The two dates are not equivalent in

kind. The fiscal year 2026 update added 487 codes, revised 38, and deleted 28 [19]; the April 1, 2026 update added, revised, and deleted none, changing instructional notes and index entries across fourteen chapters instead [20]. Procedures arrive as CPT codes revised annually by the American Medical Association and HCPCS Level II codes updated by CMS on the first business day of each quarter [21]. Laboratory observations arrive as LOINC codes, released twice yearly by the Regenstrief Institute, which announced in May 2026 a move to monthly releases targeted for March 2027 [22]. Clinical concepts arrive as SNOMED CT identifiers, with the United States Edition published each March and September even though the International Edition has moved to a monthly cycle [23]. Drug identifiers arrive as NDC codes, which change continuously as products enter and leave the market and have no release calendar at all. The claim this paper builds on is narrower than it first appears: most, not all, of these vocabularies change on dates published in advance.

The consequences of ignoring administrative change are on record, though the strongest published measurement concerns a different mechanism than is often assumed. Nestor and colleagues trained clinical prediction models on historical intensive care records and tested them on later years, finding that every model decayed. The sharpest degradation followed a system-wide record-keeping change in 2008, where a random forest lost 0.29 AUROC on mortality prediction and 0.10 on length of stay; aggregating raw features into expert-defined clinical concepts reduced those losses to 0.06 and 0.03 [3]. That event was an information system replacement rather than a vocabulary transition, and the study period predates the October 2015 move to ICD-10-CM, so it should not be cited as a measurement of the ICD-10 transition itself. It remains the clearest available demonstration of what a discrete administrative change to how care is recorded does to a deployed predictor. Davis and colleagues documented calibration drift accumulating over years of production use [4], Guo and colleagues found published approaches to temporal shift to be reactive rather than anticipatory [5], and Vela and colleagues measured degradation across a broad sample of production models [6]. The drift detection literature offers mature machinery, from the surveys of Gama and colleagues [7] to sequential schemes rooted in statistical process control [16], [17], recently applied to clinical monitoring [8], [13]. Across this work, terminology releases appear as after-the-fact explanations for anomalies rather than as scheduled events a monitor should anticipate.

The natural home for a solution is machine learning operations, which already supplies production systems with versioning discipline, registries, and monitoring hooks [12], [14], [15]. To establish what that tooling offers, we reviewed the public documentation of five widely used components in August 2026: Feast, MLflow Model Registry, Evidently, and the feature store and model registry documentation of Databricks and Amazon SageMaker. The review asked whether any lets a feature definition or registered model version declare the edition of an external controlled vocabulary it was built against, and whether any ingests a maintainer release calendar. Each versions its own artifacts capably: Feast tracks feature view schema changes and expects each model version to map to a distinct feature service, MLflow registers model versions and logs population stability index values as generic metrics, and Evidently computes drift metrics over arbitrary columns. None of the documentation reviewed described a binding between an artifact and an external terminology edition, and none treated a release calendar as an input. This is a documentation review rather than a source audit, so the claim is limited accordingly.

That absence matters because scheduled drift permits a governance posture that unscheduled drift does not. An organization cannot know when a pandemic will shift its case mix, but it knows the date on which 487 new codes become reportable, and it holds the addenda months beforehand. We first define terminology-induced drift and give it internal structure, since the mechanisms differ enough in tempo that treating them as one phenomenon yields the wrong monitoring design. We then specify TIDE, whose registry converts an undeclared data dependency into a queryable one, whose monitoring layer computes six statistics chosen so that at least one responds to each mechanism class, and whose change control process scores every deployed model against every announced release on a normalized exposure index. We map the resulting artifacts onto the transaction obligations of 45 CFR Part 162 and, for models regulated as devices, onto the predetermined change control plan pathway. Finally we run the design against a seeded synthetic stream and specify the field study that would test whether it helps.

2. BACKGROUND AND RELATED WORK

2.1 Dataset Shift in Clinical Machine Learning

Dataset shift research distinguishes covariate shift, label shift, and concept drift as formal categories [2], [7]. The clinical record is extensive. Nestor and colleagues demonstrated year-over-year degradation in intensive care models and recommended aggregating features into clinically meaningful groupings rather than modelling raw codes, a recommendation this paper returns to because the same aggregation suppresses the signal a terminology monitor needs [3]. Kansal and colleagues examined how the choice of diagnosis code grouping affects prediction performance across multiple sites, using data spanning the ICD-9-CM to ICD-10-CM transition date [24]. Otles and colleagues observed a deployed hospital model prospectively and found shift effects that retrospective evaluation had missed [9]. Saez and colleagues built tooling for delineating temporal shifts in electronic health records and showed that abrupt shifts frequently align with information system and coding changes [10]. Subbaswamy and Saria organized mitigation into reactive and proactive families and argued that proactive design is underused [11]. What is missing is the recognition that one large family of shifts is not merely predictable in form but dated in advance.

2.2 MLOps and Production Model Monitoring

The MLOps literature supplies the operational scaffolding. Sculley and colleagues identified undeclared data dependencies as a mechanism through which machine learning systems accumulate hidden technical debt, naming unstable data dependencies, of which changing vocabularies are a textbook case, as a primary risk [12]. Breck and colleagues proposed the ML Test Score, which includes tests for upstream data source change [14]. Kreuzberger and colleagues consolidated MLOps into a reference architecture with registries, feature stores, and continuous monitoring [15]. Closest to this work, Khattak and colleagues surveyed machine learning health operations and catalogued statistical process control among the monitoring families for clinical pipelines [13], and Feng and colleagues framed post-deployment monitoring as a quality improvement discipline with CUSUM-style schemes [8]. None models the vocabulary release calendar as a monitoring input.

3.3 Statistical Process Control in Healthcare

Statistical process control has six decades of healthcare application, with Benneyan and colleagues providing the canonical treatment of chart selection for healthcare processes [25]. The sequential schemes underlying modern drift detectors originate in the same tradition, in the CUSUM chart of Page [16] and the EWMA chart of Roberts [17], with Montgomery the standard reference on their design and average run length properties [26]. Healthcare attribute data routinely violate the binomial assumption behind a conventional p-chart, and Laney showed how to measure and correct the resulting overdispersion [27], [28]. The novelty available here is not the charts but their coupling to a known event calendar. Chart design trades in-control average run length, meaning how long before a false alarm, against out-of-control average run length, meaning how fast a real shift is caught [26]. A regime that cannot anticipate events must set that trade conservatively everywhere and pay for sensitivity with false alarms. A regime that knows when a special cause is scheduled can decide in advance how a signal on that date will be read. Terminology releases are that rare thing in production machine learning, a special cause published by the responsible maintainer months before it takes effect.

3.4 The Gap

Table I positions the present work. Drift detection methods detect but do not anticipate. Clinical shift studies diagnose but do not operationalize. MLOps frameworks operationalize monitoring but are calendar blind. Healthcare statistical process control brings the machinery without applying it to terminology governance. Regulatory instruments create obligations without specifying engineering practice.

TABLE I. Gap Analysis: Related Work Versus TIDE Requirements

Body of Work	Detects Drift	Anticipates Scheduled Drift	Terminology Versioning	Healthcare Governance	SPC Foundation
Concept drift detection [7], [16], [17]	Yes	No	No	No	Partial
Clinical dataset shift studies [1], [3], [9]	Diagnoses	No	No	Partial	No
MLOps frameworks [13], [14], [15]	Yes	No	No	Partial	Partial
Healthcare SPC [25], [26]	Process level	No	No	Yes	Native
45 CFR Part 162, FDA PCCP [29], [30]	No	Implied	For transactions	Yes	No
TIDE (this work)	Yes	Yes, calendar aware	Yes, registry pinned	Yes	Yes

3. A TAXONOMY OF TERMINOLOGY-INDUCED DRIFT

Terminology-induced drift is defined here as any change in the joint distribution of model inputs and outputs whose proximate cause is a change in the controlled vocabularies used to encode clinical reality, rather than a change in that reality. The April 1, 2026 ICD-10-CM update illustrates the distinction cleanly. It created no codes, retired none, and revised no descriptors. It did revise the alphabetic index so that documentation of a neuroendocrine tumor now indexes to the malignant category C7A rather than the benign D3A.8, and so that a documented diagnosis of Parkinson disease directs the coder to G20.A1 rather than the path that previously led to G20.C [20]. No patient changed, no disease changed, no code changed. What changed is which existing code the same documentation produces, and because the malignant category carries complication and comorbidity status while the benign one does not, the change propagates into severity capture and risk adjustment. A monitor keyed to code counts would have scored that release as a null event.

The scale of routine churn is easy to underestimate and equally easy to overstate, because each vocabulary publishes on its own cycle and each release differs in magnitude. Table II gives the verified ICD-10-CM record across four consecutive boundaries. October releases vary by a factor of more than two and a half in additions between fiscal years 2026 and 2027, and the April 2026 release contains no code changes at all. Exposure must therefore be computed against each specific release delta rather than assumed from a typical release.

TABLE II. ICD-10-CM Release Magnitude Across Four Consecutive Boundaries

Release	Effective Date	Added	Revised	Deleted	Note and Index Changes
FY2025 [31]	Oct. 1, 2024	252	13	36	Yes
FY2026 [19]	Oct. 1, 2025	487	38	28	Yes
April 2026 [20]	Apr. 1, 2026	0	0	0	Yes, fourteen chapters
FY2027 [32]	Oct. 1, 2026	190	4	30	Yes

Counting boundaries requires a definition, since a loose count invites double counting. We define a scheduled vocabulary boundary as a date on which a maintainer places a new edition into effect for reporting purposes, counted once per vocabulary per date and only for vocabularies the model actually consumes. A model consuming diagnoses, procedures, laboratory observations, and clinical concepts then faces eleven boundary events in a typical year: two for ICD-10-CM, one for CPT, four for HCPCS Level II, two for LOINC, and two for the SNOMED CT United States Edition. Those eleven fall on eight distinct calendar dates, because CPT and the first HCPCS quarter share January 1 while ICD-10-CM shares April 1 and October 1 with the second and fourth HCPCS quarters. A model consuming only diagnoses and professional procedures faces seven events on five dates. The count rises with ICD-10-PCS, and again for laboratory-heavy models once LOINC completes its move to monthly releases [22].

Five mechanism classes cover the observed phenomena, summarized in Table III. T1, structural replacement, is the wholesale substitution of one vocabulary for another, as in the 2015 ICD-9-CM to ICD-10-CM transition. ICD-11 came into effect globally on January 1, 2022, but no United States implementation date has been set, so any domestic transition should be treated as conditional and undated rather than imminent [33]. T2, incremental release churn, is the routine addition, deletion, and revision of codes at each release; individually small, these changes concentrate in clinically active areas, which is where model features carry signal. T3, guideline and index semantic drift, occurs when guidance changes the meaning or usage of codes that themselves remain unchanged, as the April 2026 index revisions did [20]; the code exists before and after, but its statistical relationship to the outcome moves. T4, adoption lag drift, is the gradual transition as coders, encoder software, and payer edit rules absorb a release at different speeds, producing weeks of mixed vocabulary data around each boundary. Adoption lag is partly institutionalized: the LOINC Committee has recommended adopting a new version within ninety days of release rather than on the release date [22]. T5, mapping decay, affects every pipeline that normalizes local codes through crosswalks. The clearest case is the General Equivalence Mappings, which CMS committed to updating for roughly three years after ICD-10 implementation and now maintains only as an archive [34], while ICD-10-CM has absorbed further annual updates since. Any pipeline still translating through frozen GEMs is mapping into a vocabulary that has moved.

TABLE III. Terminology-Induced Drift: Mechanism Classes, Tempo, and Observable Signatures

Class	Mechanism	Tempo	Primary Observable Signature	Representative Event
T1	Structural vocabulary replacement	Years, announced far ahead	Step collapse of known code rate; sparsity jump	ICD-9-CM to ICD-10-CM, Oct. 2015; conditional ICD-11 adoption
T2	Incremental release churn (add, delete, revise)	Semiannual (ICD-10-CM), annual (CPT), quarterly (HCPCS)	Rising unseen code rate; retired code persistence decay	FY2026 ICD-10-CM: 487 added, 38 revised, 28 deleted
T3	Guideline and index semantic drift	With each guideline or index edition	Feature-outcome association shift at stable code frequency	April 2026 index change moving neuroendocrine tumors from D3A.8 to C7A
T4	Adoption lag drift	Weeks to months after each boundary	Bimodal code usage; site and coder level divergence	Mixed old and new code streams after any release
T5	Mapping and crosswalk decay	Continuous, concentrated at releases	Unmapped code rate growth; silent feature null inflation	Frozen GEMs; local dictionary to LOINC map rot

4. THE TIDE FRAMEWORK

TIDE organizes the response into four layers: a release registry that makes vocabulary versions first-class artifacts, a calendar-aware monitoring layer, a change control and retraining process, and a governance mapping. Because the framework targets organizations that already run quality management programs, the operational structure is stated explicitly rather than by analogy. The registry phase produces a versioned release artifact and a model-to-code set compatibility matrix, owned by MLOps engineering, gated on ingestion of the delta manifest before any release enters a production window. The measurement phase produces the six chart series of Table IV, owned by MLOps engineering, gated on a completed Phase I baseline per model. The analysis phase produces a signature match decision per signal, owned jointly by health information management and the model owner, gated on a documented disposition for every in-window signal. The improvement phase produces a challenger evaluation for every model above its exposure threshold, owned by the model owner, gated on pre-specified promotion criteria. The control phase produces the release disposition record, owned by the governance committee, gated on sign-off before the window closes.

4.1 Layer 1: The Terminology Release Registry

The foundational move is to version vocabularies the way MLOps already versions models and data. For each code set edition the registry holds the effective date, the delta against the prior edition, a cryptographic hash of the source artifact, and the official mapping files where they exist. Licensing constrains what may be stored rather than what may be referenced. ICD-10-CM and HCPCS Level II files are United States government works and may be held in full. CPT is copyrighted by the American Medical Association and SNOMED CT requires a National Library of Medicine Affiliate License, so for those the registry stores metadata, code identifiers, effective dates, and hashes, resolving licensed content from the licensed source at use time rather than redistributing it. Mappings deserve equal caution, being neither uniformly available nor lossless: the March 2026 SNOMED CT United States Edition ships a map covering more than 126,000 source concepts to ICD-10-CM targets [23], while the GEMs many pipelines still rely on stopped being updated years ago [34]. Every feature definition declares the terminology versions it was built against, and every model version records those of its training data. This converts the undeclared dependency Sculley and colleagues warned about [12] into an inspectable one, so that whether a given model understands a given code becomes a registry lookup rather than a production incident. The registry also ingests the release calendar, since ICD-10-CM files publish months before their effective dates [19] and the other maintainers follow published schedules.

4.2 Layer 2: Calendar-Aware SPC Monitoring

The monitoring layer tracks a small set of terminology-sensitive statistics as control-chart series. The leading indicator is the unseen code rate, the proportion of arriving codes the deployed model vocabulary does not recognize. Treating day t as a sample of $n(t)$ coded events with $x(t)$ unrecognized, the daily proportion is charted with day-specific limits bounded to the unit interval:

$$UCL(t) = \min[1, p + L \sqrt{p(1-p)/n(t)}]$$

$$LCL(t) = \max[0, p - L \sqrt{p(1-p)/n(t)}]$$

where p is the baseline rate and L the limit multiplier. Limits are recomputed daily because claim volume varies by weekday and month. The baseline is estimated in a Phase I window of at least 90 consecutive days containing no release effective date and no known upstream system change, with signalling days removed and the estimate recomputed until the window is stable [26].

Three assumptions require attention before these limits can be trusted. Binomial independence fails in claims data because codes cluster within claims and claims within providers, inflating true variance and producing false alarms on a nominal chart; where the Phase I dispersion ratio exceeds 1.2 we substitute the Laney p -prime chart, which rescales limits by observed between-subgroup variation and reduces to the conventional chart when no overdispersion is present [27], [28]. Autocorrelation enters through weekday and holiday cycles, removed by charting day-of-week standardized residuals. Rare-event behavior matters when the expected count of unrecognized codes falls below five, where the normal approximation is unsafe and exact binomial limits are used. The unseen code rate needs no clinical risk adjustment; calibration does, and is charted within risk decile strata rather than pooled.

Slower statistics use EWMA charts, $z(t) = \lambda x(t) + (1 - \lambda) z(t-1)$ with λ set to 0.2 [17], [26]. One of them, the population stability index over grouped code frequencies, needs explicit definition because it is often reported without one. For reference proportion $e(i)$ and observed proportion $a(i)$ over bins i , PSI is the sum over bins of $(a(i) - e(i)) \ln(a(i) / e(i))$. Bins are the model grouped code categories rather than raw codes, since raw code PSI is unstable in sparse cells; bins with expected count below 30 are merged and remaining proportions floored at 0.0005 to keep the logarithm finite when a cell empties. The reference distribution is the Phase I window used for the p-chart, and PSI is computed weekly because daily values on grouped categories are dominated by weekday volume effects. PSI is non-negative with a skewed sampling distribution, so normal-theory EWMA limits do not apply; we use the EWMA purely as a smoother and set its limit at the 99.7th percentile of a bootstrap over Phase I weekly values, making the limit empirical rather than distributional. Calibration error uses a tabular CUSUM with reference value $k = 0.5$ and decision interval $h = 5$ in standardized units [16], [26].

The calendar changes how these charts are operated, and it is worth being precise about what it does not change. Chart parameters stay fixed across release windows. We considered tightening limits inside windows, which was the original design, and rejected it: tightening alters the in-control average run length, so in-window and out-of-window false alarm rates would no longer be comparable, and the evaluation in Section V.B depends on that comparison. What changes inside a window is the routing of a signal, the addition of two pre-registered delta-specific statistics, and the staffing of review. The added statistics are the arrival rate of exactly the codes on the addition list and the persistence rate of exactly those on the deletion list, both known from the registry before the window opens. A signal inside a window whose signature matches the delta confirms an anticipated transition and enters change control. A signal outside any window, or one inside whose signature does not match, indicates unscheduled drift and routes to incident response. This split is the central operational contribution, achieved through routing rather than sensitivity.

The false alarm budget is set in advance, because six charts per model across a portfolio creates a multiplicity problem informal chart design will not survive. We budget at most one expected false signal per model per quarter across the six daily charts. Over a 91 day quarter this requires a per-chart per-day false alarm probability below 0.00183. Conventional three-sigma limits give 0.0027, or 1.47 expected false signals per model-quarter, breaching the budget. Widening to $L = 3.2$ gives 0.00137, an in-control average run length near 728 days, and 0.75 expected false signals per model-quarter, which meets it. We use $L = 3.2$ throughout and report it, since a multiplier chosen after seeing the data would invalidate the evaluation. The budget governs the per-model rate; the enterprise-wide rate is addressed in Section VI.

4.3 Layer 3: Change Control and Retraining

When a release is announced, change control begins before the effective date. Impact analysis queries the registry for every deployed model whose feature vocabulary intersects the release delta and scores exposure. Let $f(c)$ be the share of the model training set coded-event mass carried by code c , normalized over coded events rather than claims so the shares sum to one even in multi-hot feature spaces. Then for model m and release r :

$$E(m, r) = w(d) S(\text{del}) + w(v) S(\text{rev}) + w(a) S(\text{add})$$

where $S(\text{del})$ is the summed $f(c)$ over deleted codes, $S(\text{rev})$ the summed $f(c)$ over revised codes each weighted by a severity term $s(c)$, and $S(\text{add})$ the summed training mass of parent categories each weighted by the fraction newly subdivided. The severity term takes 0 for a typographical revision, 0.5 for a descriptor change that narrows or broadens a concept within its parent, and 1.0 for a revision that changes the parent category or the reporting rule. It is assigned by a certified coder, and it is the one input we could not derive from the release files alone. The three weights sum to one, with defaults 0.5, 0.3, and 0.2, reflecting that a deletion is a certain loss of a known feature while an addition is a possible redistribution. Every term lies in the unit interval, so E does as well.

A worked example makes the scale concrete. Consider a denial prediction model facing the fiscal year 2026 boundary with 2.0 percent of its training mass on the 28 deleted codes, 0.8 percent on revised codes of mean severity 0.5, and 14 percent on parent categories that the 487 additions subdivide by a mean fraction of 0.10. The default weights give $E = 0.0140$, and against a tier one threshold of 0.010 this model must field a challenger. Holding the deletion weight at 0.5 and moving the addition weight from 0.10 to 0.40, with the revision weight taking the remainder, moves E only from 0.0130 to 0.0160, so every weighting in that range yields the

same disposition. The deletion term dominates, which is intended, and which means the score is most sensitive to the accuracy of the training mass estimate on deleted codes.

Exposure thresholds, set per model risk tier, decide which models watch a boundary and which must field a challenger. The registry maintains the compatibility matrix as a queryable artifact, so the organization can answer which production models have never seen the currently mandated vocabulary, a question that today typically has no owner. Exposed models enter a champion-challenger cycle in which the challenger trains on history re-expressed in the new vocabulary and both models run in parallel across the boundary under dual scoring, with promotion criteria fixed in advance. Backfilling history through a mapping is not always defensible, and we prohibit it in three situations: when the map is one-to-many and no tie-breaking rule has been documented and reviewed by a certified coder, when its provenance is a local dictionary rather than a maintainer-published artifact, and when the target concept differs from the source in a way that changes outcome ascertainment, as the April 2026 neuroendocrine tumor index change would. Where backfill is prohibited the challenger trains on post-boundary data only and promotion waits for adequate volume, a real cost the framework does not disguise.

For models regulated as devices, this loop maps onto the predetermined change control plan pathway. The applicable guidance covers AI-enabled device software functions submitted through the 510(k), De Novo, and premarket approval pathways [30]; payer claims models, risk adjustment models, and most health system operational models are not devices and fall outside it, in which case the same artifacts serve internal model risk management rather than a regulatory submission. A terminology release is close to a paradigm case of an anticipatable modification, and TIDE supplies the modification description and modification protocol content a sponsor would need. TIDE does not itself satisfy any regulatory requirement, and nothing here should be read as a claim that adopting it establishes compliance.

4.4 Layer 4: Governance

Governance closes the loop with obligations and roles, stated at their actual legal width. The HIPAA administrative simplification regulations at 45 CFR Part 162 designate the code sets adopted for covered electronic transactions and require covered entities to use the code set valid at the date of service when conducting those transactions [29]. They do not require any particular internal feature store, model registry, monitoring scheme, or retraining practice. The controls proposed here are governance recommendations rather than regulatory requirements, and their value is that they make an existing transaction obligation traceable into a model estate that would otherwise hold no record of which vocabulary each model understands. For organizations aligning to the NIST AI Risk Management Framework, the registry and compatibility matrix populate the MAP function while the chart series and disposition records populate MEASURE [35]. Health information management and coding leadership own release interpretation, since they already track these calendars for revenue cycle reasons; MLOps engineering owns the registry and monitoring layers; model owners own promotion decisions; and a standing committee owns the disposition record.

4.5 Implementation Status

A reference implementation is under development and is not yet released, so the design is specified here at the level of interfaces and parameters so that it can be rebuilt independently. The registry is a versioned object store keyed by vocabulary, edition, and effective date, populated by ingestion jobs watching the CMS, NCHS, AMA, Regenstrief, and NLM release channels. Feature definitions carry a terminology version manifest and the model registry extends its metadata with the code set editions of each training snapshot. The monitoring layer runs as a stream processor computing the Table IV statistics per model per day, persisting chart state and evaluating window posture from the event calendar. Alert routing separates the two response paths into queues with different service level expectations. Nothing requires unusual infrastructure; the design goal was that a team already running a conventional MLOps pipeline can add TIDE as a layer rather than a replacement.

TABLE IV. TIDE Monitoring Statistics, Control Schemes, and Response Paths

Monitored Statistic	Chart	Drift Classes	In-Window Response	Out-of-Window Response
Unseen code rate (daily)	p-chart	T1, T2, T4	Confirm against delta; begin challenger scoring	Incident: unregistered vocabulary change
Retired code persistence	p-chart	T2, T4	Track adoption lag; extend dual coding	Investigate upstream system lag
Feature null and unmapped rate	EWMA	T5	Refresh mappings from release files	Audit crosswalk maintenance
Grouped code frequency PSI (weekly)	EWMA	T2, T3	Compare against pre-registered signature	Case mix or documentation shift review
Model score mean and dispersion	EWMA	T2, T3, T4	Champion-challenger comparison	Full drift investigation
Calibration error (windowed)	CUSUM	T3, T4	Recalibrate or promote challenger	Model risk escalation

5. DEMONSTRATION AND EVALUATION PROTOCOL

5.1 A. Synthetic Demonstration of Detection Mechanics

To show how the six statistics behave relative to one another without overclaiming, we constructed a synthetic scenario and label it as such. No claim in this subsection derives from production healthcare data, and no figure here estimates field performance.

The generator produces 50,000 claim lines per day over 180 pre-boundary and 120 post-boundary days. Diagnosis code frequencies follow a Zipf distribution with exponent 1.1 over 2,000 baseline codes, randomly permuted per replication, reproducing the heavy frequency skew of real claim streams without copying any real distribution. Each code carries a denial risk coefficient drawn from a normal distribution with standard deviation 0.5 around an intercept of -2.0, giving a base denial rate near 12 percent. A background unseen code rate of 0.002 is present throughout. At the boundary, 487 additions subdividing high-frequency parents, 38 revisions, and 28 deletions are applied, with new codes phased in linearly over 42 days to a peak of 3.1 percent of daily code volume and deleted codes retired on the complementary schedule. The deployed model consumes one-hot features over the baseline codes only, so a new code falls through to an all-zero vector and loses its risk signal while the ground truth retains it. Phase I uses the first 90 days. All charts use $L = 3.2$, $\lambda = 0.2$, CUSUM $k = 0.5$ and $h = 5$, and a 21 day label delay is imposed on the calibration chart to represent adjudication lag. The scenario ran for 200 independent replications with seeds 1 through 200.

TABLE V. Synthetic Demonstration: Days to First Signal Across 200 Seeded Replications (Illustrative Only)

Monitored Statistic	Chart	Median Days	IQR	Signaled Within 120 Days	Signature Match
Unseen code rate	p-chart	1	1 to 2	100%	Yes, addition list
Feature null and unmapped rate	EWMA	2	2 to 2	100%	Yes, unmapped additions
Retired code persistence	p-chart	9	7 to 10	100%	Yes, deletion list
Grouped code frequency PSI	EWMA	22	15 to 22	100%	Partial
Model score mean	EWMA	32	22 to 46	84.0%	Not applicable
Windowed calibration error	CUSUM	63	55 to 69	4.5%	Not applicable

Two features of these results are built into the generator and one is not, and separating them is necessary. The speed of the unseen code rate chart is a construction: its statistic derives from the release delta, so it must respond as soon as new codes arrive, and its median of one day confirms only that the arithmetic works. The retired code persistence decay likewise follows the adoption schedule the generator imposes. What was not built in is the ordering and separation of the remaining charts. The grouped PSI reached its limit at a median of 22 days, the score mean EWMA at 32 days and in only 84 percent of replications, and the calibration CUSUM signaled within the observation period in just 4.5 percent of replications despite a 63 day median among those that did. That last result is the most useful and it is a negative finding. Outcome-based monitoring, which is what many production teams rely on, was close to blind to this event at this effect size and label delay, because a 3.1 percent volume shift into a null feature moves aggregate calibration by less than the day-to-day noise the CUSUM was scaled against. The grouped PSI is slow for a structural reason worth naming: new codes usually subdivide an existing parent, so a grouping that aggregates to parent categories absorbs precisely the change the monitor is trying to see. Feature aggregation of the kind Nestor and colleagues recommend for robustness [3] therefore trades against terminology observability, and a team adopting one should know it is giving up the other. These figures describe one generator under one set of assumptions and are not effect size estimates.

5.2 Multi-Site Pilot Evaluation Protocol

The framework has not been tested in production, and the following protocol is specified as a feasibility and pilot study rather than a confirmatory trial, for reasons the power calculation makes plain. Setting: three or more organizations running production machine learning on coded data, including at least one payer with claims models and one provider with EHR-based clinical models. Period: one year spanning two ICD-10-CM effective dates and one CPT edition. Design: within-organization comparison of TIDE against the incumbent stack running in parallel on the same model portfolio. Pre-registered outcomes: time from effective date to first true signal per exposed model; the proportion of release-related degradations detected before a downstream business detection such as a denial spike; false alarm rate per chart per month; and post-boundary recovery time with and without pre-boundary challenger preparation. Chart parameters are frozen before the first observed boundary and reported with the results, since mid-study tuning would invalidate the false alarm comparison.

Sample size is governed by boundaries rather than patients, and the arithmetic is unfavorable enough to state plainly. Three sites observing three boundaries each yield nine paired

observations. An exact two-sided sign test at the 0.05 level on nine pairs rejects only when at least eight favor TIDE. That design has 77.5 percent power if TIDE genuinely detects first on 90 percent of boundaries, and 43.6 percent power if it detects first on 80 percent. Reaching 80 percent power against the 80 percent alternative requires roughly twenty paired boundaries, implying about seven sites at three boundaries each or a two year window. Nine pairs also sit within three organizations, so the observations are clustered and a simple sign test does not account for that. We therefore specify the three-site design as a pilot whose purpose is to establish feasibility, measure the comparison logistics, and produce the variance estimates a confirmatory study would need, with per-site effects reported alongside any pooled estimate and a stratified permutation test used in place of a pooled sign test. Presenting nine clustered pairs as confirmatory evidence would misrepresent what the design supports.

6. DISCUSSION

6.1 Contribution and Implications

The core claim is conceptual before it is technical: a substantial class of healthcare machine learning drift is scheduled, and scheduled drift deserves a different engineering and governance posture than the surprise-oriented one the field inherited from general-purpose drift detection. Building the framework surfaced three trade-offs the abstract statement of that claim conceals. The first is the decision not to tighten control limits inside release windows even though doing so would improve in-window sensitivity, because a moving limit makes in-window and out-of-window false alarm rates incomparable and destroys the only clean evaluation the design admits. The second is the severity weight in the exposure score, which resisted every attempt to derive it from the release files and still requires a certified coder, leaving a human judgment inside an otherwise automatable pipeline. The third is the grouping choice for the PSI chart, where the demonstration showed that the aggregation making models robust to terminology change is the same aggregation making that change invisible to the monitor.

It is equally worth stating where TIDE will not help. It offers nothing to models consuming no coded data, such as imaging or waveform models, and only indirect coverage of models reaching terminology through a natural language normalization layer. It is defeated by an upstream vendor that silently remaps new codes to old ones before the model sees them, since the unseen code rate then never rises and the leading indicator is suppressed while semantic drift proceeds unobserved; organizations should verify what their encoder and clearinghouse layers do before trusting the chart. And it will misroute genuine clinical change that coincides with a release window, because the routing logic reads a within-window signature match as confirmation of an anticipated transition. The out-of-window paths in Table IV exist precisely because clinical reality still changes without filing advance notice.

For payers, the immediate application is claims model continuity across the ICD-10-CM cycle, where undetected T2 and T4 drift surfaces as authorization errors, payment errors, and misfiring fraud edits. Risk adjustment deserves particular attention, since hierarchical condition category models translate diagnosis codes into capitated revenue, and the April 2026 index revision moving neuroendocrine tumors into a complication and comorbidity category is exactly the kind of change that propagates within months into risk scores, plan payments, and audit exposure without any code having changed. A Medicaid managed care organization running denial prediction, care gap identification, and risk stratification meets every boundary three times over, once per model family. For health systems the application is clinical model stability; for vendors, terminology version pinning is a feature store capability gap this paper names explicitly. Feng and colleagues called for treating clinical AI monitoring as continuous quality improvement [8], and TIDE supplies the special cause taxonomy that quality improvement methodology has always required, since a control chart is only as useful as the causal vocabulary behind it [25].

The framework connects to enterprise risk management in a way worth specifying rather than gesturing at. Terminology currency is an operational risk with a known event horizon, which makes it unusually tractable under an ISO 31000 program [36]: risk identification draws the event and its date from the release calendar, exposed assets come from the compatibility matrix, controls come from Table IV, and the review cadence follows the release cadence. Residual risk should not be measured by false-negative history alone, which is unobservable without ground truth. We instead define it as the product of event likelihood, business impact, and the complements of detection effectiveness and control effectiveness, each estimated with

an explicit uncertainty range, treating boundary-attributable incidents discovered by downstream business detection after a window closed as an observable lower bound on missed events rather than the measure itself.

6.2 Limitations

Nine limitations bound the claims. First, the framework has not been validated in production, and Section V.B specifies the protocol precisely because a synthetic demonstration cannot substitute for field evidence. Second, scope is limited to structured coded data. Third, the treatment is United States-centric, though the logic transfers wherever code sets are administered on published schedules. Fourth, TIDE governs terminology-induced drift specifically and complements rather than replaces general drift monitoring. Fifth, the exposure score treats codes independently and will underestimate impact when a release restructures relationships among codes a model exploits jointly; interaction-aware scoring is an open problem the taxonomy makes visible without solving. Sixth, label delay bounds what outcome-based charts contribute, since calibration signals depend on adjudicated outcomes arriving weeks to months after the encounter, which is why the demonstration found the calibration CUSUM nearly silent and why it cannot serve as a primary in-window detector. Seventh, chart multiplicity is controlled only at the model level; the budget in Section IV.B holds the per-model rate near 0.75 signals per quarter, but an estate running hundreds of models will still generate an enterprise-wide volume needing a triage layer this paper does not design. Eighth, terminology licensing restricts what any shared registry artifact can contain, limiting reproducibility of registry contents even when the code is open. Ninth, mapping error propagates: crosswalks are incomplete and sometimes one-to-many, and any history backfilled through them inherits errors that are difficult to audit afterward.

6.3 Future Work

Beyond the evaluation protocol, three directions follow. Automated signature generation could learn expected post-release uptake curves from historical releases and sharpen the pre-registered signatures beyond simple delta membership, which matters more as calendars densify: once LOINC completes its move to monthly releases, manual per-release review stops being feasible for laboratory-heavy pipelines [22]. Cross-vocabulary interaction effects, such as an ICD-10-CM update landing on the same date as an HCPCS quarter, deserve joint monitoring designs, and the eight-date calendar derived in Section III shows such collisions are the rule rather than the exception. Should the United States eventually adopt ICD-11, that transition would be the largest T1 event since 2015, and organizations that have institutionalized this governance for routine churn would meet it with rehearsed machinery.

7. CONCLUSION

Healthcare machine learning inherits its inputs from vocabularies revised on schedules published in advance, and the field drift tooling has been ignoring the schedule. This paper defined terminology-induced drift, organized it into five mechanism classes, and specified TIDE, which versions terminology releases as MLOps artifacts, monitors six terminology-sensitive statistics on charts whose parameters remain fixed across release windows, routes anticipated and unanticipated signals down separate paths, and pins model versions to code set versions under a change control process aligned with 45 CFR Part 162 and, where applicable, the predetermined change control plan pathway. What has been demonstrated is mechanical: on a seeded synthetic stream, a statistic constructed from the release delta responds within a day while an outcome-based calibration chart signaled within four months in fewer than one run in twenty. What remains hypothetical is everything that matters operationally, including whether earlier detection changes any business outcome, whether the false alarm budget survives contact with real claim volume, and whether the routing split holds when clinical and scheduled change coincide. The next step is the pilot in Section V.B, and the analysis there is explicit that three sites and nine boundaries can establish feasibility but cannot settle the question.

References

1. S. G. Finlayson et al., "The clinician and dataset shift in artificial intelligence," *New England Journal of Medicine*, vol. 385, no. 3, pp. 283-286, 2021, doi: 10.1056/NEJMc2104626.

2. J. Quinonero-Candela et al., Eds., *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.
3. B. Nestor et al., "Feature robustness in non-stationary health records: caveats to deployable model performance in common clinical machine learning tasks," in *Proc. 4th Machine Learning for Healthcare Conf.*, PMLR vol. 106, 2019, pp. 381-405.
4. S. E. Davis, T. A. Lasko, G. Chen, E. D. Siew, and M. E. Matheny, "Calibration drift in regression and machine learning models for acute kidney injury," *J. American Medical Informatics Association*, vol. 24, no. 6, pp. 1052-1061, 2017, doi: 10.1093/jamia/ocx030.
5. L. L. Guo et al., "Systematic review of approaches to preserve machine learning performance in the presence of temporal dataset shift in clinical medicine," *Applied Clinical Informatics*, vol. 12, no. 4, pp. 808-815, 2021, doi: 10.1055/s-0041-1735184.
6. D. Vela et al., "Temporal quality degradation in AI models," *Scientific Reports*, vol. 12, art. 11654, 2022, doi: 10.1038/s41598-022-15245-z.
7. J. Gama et al., "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, art. 44, 2014, doi: 10.1145/2523813.
8. J. Feng et al., "Clinical artificial intelligence quality improvement: towards continual monitoring and updating of AI algorithms in healthcare," *npj Digital Medicine*, vol. 5, art. 66, 2022, doi: 10.1038/s41746-022-00611-y.
9. E. Otles et al., "Mind the performance gap: examining dataset shift during prospective validation," in *Proc. 6th Machine Learning for Healthcare Conf.*, PMLR vol. 149, 2021, pp. 506-534.
10. C. Saez et al., "EHRtemporalVariability: delineating temporal data-set shifts in electronic health records," *GigaScience*, vol. 9, no. 8, gaaa079, 2020, doi: 10.1093/gigascience/giaa079.
11. A. Subbaswamy and S. Saria, "From development to deployment: dataset shift, causality, and shift-stable models in health AI," *Biostatistics*, vol. 21, no. 2, pp. 345-352, 2020, doi: 10.1093/biostatistics/kxz041.
12. D. Sculley et al., "Hidden technical debt in machine learning systems," in *Proc. 28th Int. Conf. Neural Information Processing Systems*, Montreal, QC, Canada, 2015, pp. 2503-2511.
13. F. K. Khattak et al., "MLHops: machine learning health operations," *IEEE Access*, vol. 12, pp. 92161-92180, 2024, doi: 10.1109/ACCESS.2024.3422367.
14. E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "The ML test score: a rubric for ML production readiness and technical debt reduction," in *Proc. IEEE Int. Conf. Big Data*, Boston, MA, USA, 2017, pp. 1123-1132, doi: 10.1109/BigData.2017.8258038.
15. D. Kreuzberger, N. Kuehl, and S. Hirschl, "Machine learning operations (MLOps): overview, definition, and architecture," *IEEE Access*, vol. 11, pp. 31866-31879, 2023, doi: 10.1109/ACCESS.2023.3262138.
16. E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1-2, pp. 100-115, 1954, doi: 10.1093/biomet/41.1-2.100.
17. S. W. Roberts, "Control chart tests based on geometric moving averages," *Technometrics*, vol. 1, no. 3, pp. 239-250, 1959, doi: 10.1080/00401706.1959.10489860.
18. Centers for Medicare and Medicaid Services and National Center for Health Statistics, "ICD-10-CM official guidelines for coding and reporting, FY 2026," 2025. [Online]. Available: <https://www.cms.gov/files/document/fy-2026-icd-10-cm-coding-guidelines.pdf>. [Accessed: Aug. 30, 2026].
19. National Center for Health Statistics, "ICD-10-CM files, FY 2026 update, effective October 1, 2025," Hyattsville, MD, USA, 2025. [Online]. Available: <https://www.cdc.gov/nchs/icd/icd-10-cm/files.html>. [Accessed: Aug. 30, 2026].
20. Centers for Medicare and Medicaid Services, "ICD-10-CM April 1, 2026 update: instructional note and index revisions," Baltimore, MD, USA, 2026. [Online]. Available: <https://www.cms.gov/medicare/coding-billing/icd-10-codes>. [Accessed: Aug. 30, 2026].
21. Centers for Medicare and Medicaid Services, "HCPCS quarterly update," Baltimore, MD, USA. [Online]. Available: <https://www.cms.gov/medicare/coding-billing/healthcare-common-procedure-system/quarterly-update>. [Accessed: Aug. 30, 2026].
22. Regenstrief Institute, "Evolving LOINC release cadence for more timely updates," Indianapolis, IN, USA, May 4, 2026. [Online]. Available: <https://loinc.org/monthly>. [Accessed: Aug. 30, 2026].
23. National Library of Medicine, "SNOMED CT United States Edition, March 2026 release, version 20260301," Bethesda, MD, USA, 2026. [Online]. Available: https://www.nlm.nih.gov/healthit/snomedct/us_edition.html. [Accessed: Aug. 30, 2026].
24. A. Kansal et al., "Impact of diagnosis code grouping method on clinical prediction model

- performance: a multi-site retrospective observational study," *Int. J. Medical Informatics*, vol. 151, art. 104466, 2021, doi: 10.1016/j.ijmedinf.2021.104466.
25. J. C. Benneyan, R. C. Lloyd, and P. E. Plsek, "Statistical process control as a tool for research and healthcare improvement," *Quality and Safety in Health Care*, vol. 12, no. 6, pp. 458-464, 2003, doi: 10.1136/qhc.12.6.458.
26. D. C. Montgomery, *Introduction to Statistical Quality Control*, 8th ed. Hoboken, NJ, USA: Wiley, 2020.
27. D. B. Laney, "Improved control charts for attributes," *Quality Engineering*, vol. 14, no. 4, pp. 531-537, 2002, doi: 10.1081/QEN-120003555.
28. M. A. Mohammed and D. Laney, "Overdispersion in health care performance data: Laney approach," *Quality and Safety in Health Care*, vol. 15, no. 5, pp. 383-384, 2006, doi: 10.1136/qshc.2006.017830.
29. U.S. Department of Health and Human Services, "HIPAA administrative simplification: standards for electronic transactions and code sets," 45 CFR Part 162, as amended. [Online]. Available: <https://www.ecfr.gov/current/title-45/part-162>. [Accessed: Aug. 30, 2026].
30. U.S. Food and Drug Administration, "Marketing submission recommendations for a predetermined change control plan for artificial intelligence-enabled device software functions," Docket FDA-2022-D-2628, Silver Spring, MD, USA, Aug. 2025; final guidance first issued Dec. 2024. [Online]. Available: <https://www.fda.gov/regulatory-information/search-fda-guidance-documents>. [Accessed: Aug. 30, 2026].
31. Centers for Medicare and Medicaid Services, "ICD-10-CM files, FY 2025 update, effective October 1, 2024," Baltimore, MD, USA, 2024. [Online]. Available: <https://www.cms.gov/medicare/coding-billing/icd-10-codes>. [Accessed: Aug. 30, 2026].
32. Centers for Medicare and Medicaid Services and National Center for Health Statistics, "ICD-10-CM files, FY 2027 update, effective October 1, 2026," 2026. [Online]. Available: <https://www.cms.gov/medicare/coding-billing/icd-10-codes>. [Accessed: Aug. 30, 2026].
33. World Health Organization, "ICD-11 implementation: frequently asked questions," Geneva, Switzerland, 2025. [Online]. Available: <https://www.who.int/standards/classifications/frequently-asked-questions/icd-11-implementation>. [Accessed: Aug. 30, 2026].
34. Centers for Medicare and Medicaid Services, "ICD-10-CM and ICD-10-PCS general equivalence mappings archive," Baltimore, MD, USA. [Online]. Available: <https://www.cms.gov/medicare/coding-billing/icd-10-codes/icd-10-cm-icd-10-pcs-gem-archive>. [Accessed: Aug. 30, 2026].
35. National Institute of Standards and Technology, "Artificial intelligence risk management framework (AI RMF 1.0)," NIST AI 100-1, Gaithersburg, MD, USA, Jan. 2023, doi: 10.6028/NIST.AI.100-1.
36. International Organization for Standardization, "Risk management: guidelines," ISO 31000:2018, Geneva, Switzerland, 2018.