Designing Safety Specifications for Clinical AI: A Case Study

Shibbir Ahmed

Department of Computer Science
Texas State University
shibbir@txstate.edu

Abstract

Clinical AI models increasingly inform care decisions, yet implicit assumptions about data timing, label semantics, calibration, and operating thresholds are rarely specified or monitored, causing subtle failures with standard metrics. We present executable safety contracts, lightweight, task-level specifications enforced as runtime checks for hospital length-of-stay prediction. The specifications capture preconditions (data integrity, index-time alignment, censoring), postconditions (admissible outputs, alert-budget bounds), and invariants (coverage/calibration targets, subgroup equity). We implement these checks in a Python pipeline and evaluate them on a single-center MIMIC-IV cohort and a multi-center eICU-style cohort using simple baselines (logistic regression, gradient boosting) with conformal intervals and post-hoc calibration. The contracts exposed hazards that MAE (Mean Absolute Error), AUC (Area Under the ROC Curve), or ECE (Expected Calibration Error) alone missed, for example, acceptable point error with severe under-coverage in eICU, well-calibrated probabilities that nonetheless violated alert-rate constraints, and dataset-specific fairness gaps. Lightweight remedies such as conformal radius tuning, threshold/alert-scope selection, and calibration often restored compliance without degrading point performance, while clarifying when deeper modeling or policy changes were needed. Overall, the case study shows that Design by Contract principles extend beyond APIs to system-level specifications for clinical ML, providing a practical way to state safety expectations, check them with minimal compute, and make violations actionable.

Introduction

Clinical AI systems now play a growing role in care delivery, yet questions about their safety persist. Unlike conventional software APIs, where contracts make expected behavior explicit (Meyer 1992), clinical prediction models are often sent to production with a bundle of undocumented assumptions about inputs, label construction, operating thresholds, and intended use (Chen et al. 2020; Kapoor and Narayanan 2023). Those assumptions are easy to miss and drift; nothing necessarily crashes when they fail. A model may still clear the usual benchmarks while behaving unsafely in practice, for instance, appearing "well calibrated" yet leading to poor decisions at realistic thresholds (Guo et al. 2017). These assumptions can be subtle, evolve over time, and, when violated, may not trigger explicit failure. As a result, models may pass standard benchmarks while still behaving in ways that jeopardize patient safety (Guo et al. 2017).

One solution, drawn from software engineering, is Design by Contract (DbC), a notion where each component declares what it requires to run, what it promises in return, and what conditions must always hold (Meyer 1992). Introduced initially to make software more predictable, these ideas can be reinterpreted for machine learning systems, especially as recent work has shown that many deep learning bugs arise from unspoken assumptions about data, training procedures, or architecture choices (Ahmed et al. 2023; Humbatova et al. 2020; Islam et al. 2019).

This paper brings that perspective to hospital length-of-stay (LOS) prediction, where seemingly minor specification slips can have outsized clinical impact. We encode expectations about timing alignment and leakage control, label integrity and censoring, probability calibration, distribution-free uncertainty, operational alert burden, and subgroup equity as small, executable checks that wrap data assembly, training, and inference, adapting the DL-contract (Ahmed et al. 2023) to the clinical context. We evaluate these contracts in a Python pipeline across two ICU cohorts with different sources of variability: a single-center dataset based on MIMIC-IV (Johnson et al. 2023) and a multi-center eICU-style cohort (Pollard et al. 2018).

For this objective, we ask three concrete research questions. Do contract checks identify safety-relevant failures that metrics miss, for example, a model with good MAE but inadequate coverage, or well-calibrated probabilities that still blow past an alert budget (Guo et al. 2017)? How portable are passes and violations across cohorts that differ in case mix and practice patterns (single- vs. multi-center)? Can simple adjustments, like conformal tuning or threshold selection post-calibration, restore compliance without degrading point performance (Angelopoulos and Bates 2023)? As our results show, the contracts make these questions testable at low cost and, in several cases, reveal hazards that would not be apparent from accuracy or AUC alone.

Background

Design by Contract Design by Contract (DbC) views software components as parties to an agreement: a routine declares what it requires to start, what it guarantees on completion, and what must remain true throughout execution. Meyer's formulation in the Eiffel language made these ideas concrete via preconditions, postconditions, and invariants (Meyer 1992). Beyond runtime checks, DbC functions as precise documentation and a foundation for reasoning about correctness, useful whenever silent failures carry real cost. DL Contract (Ahmed et al. 2023) is an innovative adaptation of design-by-contract for deep learning libraries. It allows developers to define key properties of architectures, data, and training processes, streamlining the detection of bugs in deep learning software while enhancing reliability with minimal overhead

How we can apply DbC to clinical prediction models This case study paper utilizes DbC as a terminology for *safety specifications* rather than a programming technique. Concretely, we map:

- **Preconditions**: constraints that must hold *before* a prediction is produced (e.g., input schema and units are valid; timestamps are aligned so no information recorded after the prediction point is used; censoring is handled consistently).
- **Postconditions**: guarantees about the *outputs* (e.g., scores lie in admissible ranges; alerting policies are triggered only within defined operating regions).
- **Invariants**: properties that should persist *over time and cohorts* (e.g., calibration error stays below a threshold; alert rate remains within staffing limits; data integrity checks continue to pass after updates).

We use the term *contract* to denote a triple $(scope, check, action)$: where in the pipeline it applies, how it is verified, and what happens on violation (log, block, or degrade gracefully). This section provides the conceptual part only; detailed connections to prior work and existing systems are discussed in the Related Work.

Related Work

Ensuring safety in machine learning has long been a key focus in software engineering and AI. One practical approach is using formal contracts or specifications to clarify guarantees. Seshia et al. (Seshia et al. 2018) highlight that rigorous specifications are vital for trustworthy AI, especially in high-stakes situations. At the same time, subsequent work has explored how testing and contracts can be combined to reveal faults that would otherwise remain hidden (Mens, Decan, and Spanoudakis 2019; Riccio et al. 2020).

At the same time, the rise of deep learning has created new safety concerns that traditional software verification cannot easily capture. Studies by Zhang et al. (Zhang et al. 2018, 2020) show that neural networks are brittle under distributional shifts and adversarial input. Attempts to repair or adapt faulty models after deployment—such as approaches by Wan et al. (Wan et al. 2021) highlight the difficulty of reconciling accuracy with reliability once a model is in use. Beyond repair, research on debugging (Wardat, Le, and Rajan 2021; Wardat et al. 2022) and interpretability (Liu et al. 2021; Cao et al. 2022) illustrates how contract-like reasoning can help trace model behavior, but also underscores how incomplete these safeguards remain.

Practical tools have brought contract principles into programming environments. Linters and analyzers such as `Pylint` (PyCQA 2016), `PyTA` (Lorena Buciu and et al. 2016), and `PyContract` (Graham et al. 2010) embody lightweight forms of design-by-contract for code. More recent work seeks to extend such ideas into machine learning pipelines, for instance by automating compliance checks (Khairunnesa et al. 2023; Nikanjam et al. 2021) or embedding safety checks into training frameworks like Ariadne (Dolby et al. 2018) and AutoTrainer (Zhang et al. 2021). These efforts indicate growing recognition that contracts must operate at multiple levels, from code to models to systems. Recent work (Ahmed et al. 2023) has introduced DL Contract, an adaptation of design-by-contract for deep learning libraries that specifies properties of architectures, data, and training processes, enabling developers to catch and diagnose bugs in DL software with low overhead and demonstrated effectiveness.

In healthcare specifically, the literature points to recurring hazards arising less from obvious bugs than from subtle specification gaps. Chen et al. (Chen et al. 2020) emphasize that biases in data collection, poorly aligned labels, and miscalibrated predictions can compromise safety without producing visible errors. Similar concerns appear in taxonomies of failure modes for clinical prediction (Lagouvardos et al. 2020; Humbatova et al. 2020; Islam et al. 2019, 2020), where issues like temporal leakage, censoring, and operational drift routinely emerge. Broader surveys of medical AI safety argue that while performance benchmarking is common, systematic specification of "safe" use is still underdeveloped.

Recent policy and governance initiatives reflect this gap. National guidelines, including the U.S. Executive Order on AI safety, call for practices such as incident tracking, stress testing, and traceability, yet offer little guidance on how such requirements translate into concrete specifications for predictive models. Reviews of AI in risk management and patient safety reach a similar conclusion: the promise of AI is evident, but standards for specification and monitoring remain inconsistent. Technical surveys add to the picture, highlighting explainability, bias, and workflow integration as persistent barriers to specifying safe clinical AI. Frameworks for "trustworthy AI" and functional requirements catalogues offer starting points, but they fall short of the kind of domain-tailored contracts needed in practice.

Our work responds directly to these shortcomings. We propose explicit, system-level safety contracts for clinical prediction models that capture requirements on data integrity, timing, label definition, calibration, and operational thresholds. In doing so, we connect design-by-contract principles from software engineering with the pressing need for practical, verifiable specifications in healthcare AI.

Study Design

Objectives and Research Questions

We evaluate whether *executable contracts*, lightweight, task-level specifications enforced as runtime checks, improve the safety and operational suitability of hospital length-of-stay (LOS) prediction. Experiments use a single-center cohort (MIMIC-IV) and a multi-center cohort (eICU). We focus on three questions:

- **RQ1 (Detection):** Do contract checks (coverage for regression, alert-budget for classification, subgroup equity) flag safety-relevant failures that point metrics (e.g., MAE/AUC) or ECE alone do not?
- **RQ2 (Portability):** How stable are contract outcomes across cohorts, single-center vs. multi-center, and where do passes/violations flip when data distribution and practice patterns change?
- **RQ3 (Remediation):** Can light-weight, principled adjustments (e.g., conformal radius tuning, post-hoc probability calibration, threshold/alert-scope selection) restore contract compliance without degrading accuracy?

Operational definitions. Unless noted otherwise, we target: (i) regression coverage 0.80 ± 0.10; (ii) classification alert rate within a 10–40% operational band; and (iii) subgroup error gap bounded by $\max(6\text{ h}, 0.5 \times \text{MAE})$. These thresholds make each research question falsifiable and align results with clinical constraints.

Datasets, Cohorts, and Baselines

Cohorts. We evaluate two ICU cohorts constructed according to a common schema. First, a single-center cohort derived from the publicly released MIMIC-IV materials (Johnson et al. 2023). Second, a multi-center eICU-style cohort following the eICU-CRD schema (Pollard et al. 2018); when the full CSV export is unavailable, we generate a synthetic eICU folder with matched column layout and clinically plausible marginals (Section).

Common cohort schema. For both cohorts, we define a *6-hour* index window after ICU admission and assemble a wide table with `stay_id`, `subject_id`, `index_time`, `los_hours`, `censored`, `age`, `lactate_max_0--6h`, `is_micu`, `is_sicu`, `first_careunit`, plus timestamp columns used by our temporal contracts. Labels are (i) *regression*: `los_hours` (continuous); and (ii) *classification*: $\mathbb{1}[\texttt{los\_hours} \geq 48]$. We apply patient-level splits (60/20/20 train/validation/test; fixed seeds) to prevent identity leakage across splits, consistent with dataset conventions and leakage guidance (Johnson et al. 2023; Pollard et al. 2018; Kapoor and Narayanan 2023).

Baselines and training setup. To isolate the value of contracts rather than model complexity, we use simple, widely available baselines implemented in SCIKIT-LEARN (Pedregosa et al. 2011). For *regression* (LOS in hours), we train a Gradient Boosting Regressor (Friedman 2001) to the conditional median (quantile $\alpha = 0.5$) and wrap it with split-conformal prediction to obtain distribution-free 80% intervals (Lei et al. 2018; Angelopoulos and Bates 2023). For *classification* (LOS $\geq$ 48h), we use Logistic Regression as a GLM (McCullagh and Nelder 1989) and a Gradient Boosting Classifier (Friedman 2001); both are post-hoc calibrated via 3-fold *isotonic* (Zadrozny and Elkan 2002) and *Platt* (sigmoid) calibration (Platt 1999), following calibration best practices (Guo et al. 2017). Where noted, we optionally substitute LightGBM's quantile objective for fast quantile regression (Ke et al. 2017). Hyperparameters are modest (GBR with default depth, 200–400 estimators; Logistic Regression with `lbfgs`, `max_iter`=2000) to keep runs reproducible and to focus on specification.

Contract Collection

The contracts in our suite are drawn from three sources: the *mechanism* of executable specifications from Design by Contract for deep learning APIs (Ahmed et al. 2023); (i) *cohort construction and temporal modeling* practices in EHR ML (Johnson et al. 2023; Pollard et al. 2018; Shickel et al. 2018; Kapoor and Narayanan 2023; Quionero-Candela et al. 2009); and (ii) *clinical operations* evidence on alert burden and decision support (van der Sijs et al. 2006; Ancker et al. 2017). Table 1 summarizes each contract, its rationale, what is checked, and key sources.

Cohort & causality. **Index-time alignment** prevents temporal leakage, a well-documented cause of overly optimistic performance (Kapoor and Narayanan 2023; Shickel et al. 2018). **Label integrity & censoring** treats LOS as a time-to-event quantity derived from admission and discharge times, with explicit censor flags when stays are incomplete (Johnson et al. 2023; Pollard et al. 2018). **Patient-level splitting** avoids identity leakage across train/validation/test, matching MIMIC-IV and eICU schema conventions (Johnson et al. 2023; Pollard et al. 2018). **Bounded extrapolation** detects validation feature values far outside the training envelope, providing an early warning for distribution shift (Quionero-Candela et al. 2009).

Statistical robustness. **Conformal coverage** applies split-conformal prediction to yield distribution-free intervals with finite-sample guarantees; in our study, we target 80% as a conservative default, though the method supports any level (Lei et al. 2018; Angelopoulos and Bates 2023). **Calibration (ECE)** checks probability reliability; isotonic and Platt calibration remain standard and effective post-hoc fixes (Guo et al. 2017; Zadrozny and Elkan 2002; Platt 1999). **Temporal consistency** enforces a clinically plausible stability condition: after a benign improvement (e.g., lactate -0.5), predicted remaining LOS should not increase implausibly. This aligns with temporal modeling practice in EHRs (Shickel et al. 2018). **Fairness (group gap)** compares MAE across care units and constrains disparities, reflecting safety concerns in clinical AI (Obermeyer et al. 2019).

Operational/deployment. **Alert-budget band** (10–40%) constrains alert rates to align with staffing capacity, mitigating alert fatigue and overrides (van der Sijs et al. 2006; Ancker et al. 2017). **Occupancy plausibility** verifies that aggregate predicted LOS remains within historical

Contract	Rationale	What we check (summary)	Key sources
Index-time alignment (no leakage)	Prevent look-ahead bias from using data recorded after the prediction point	For each feature with a timestamp: enforce `feature_time` $\leq$ `index_time` row-wise	(Kapoor and Narayanan 2023; Shickel et al. 2018)
Label integrity & censoring	LOS must be derived consistently from admit/discharge times; censoring matters when stays are incomplete	LOS = discharge$-$admit (non-negative); censored rows must have a valid flag	(Johnson et al. 2023; Pollard et al. 2018)
Patient-level splits	Avoid identity leakage and overly optimistic estimates	Ensure disjoint `subject_id` across train/val/test splits	(Johnson et al. 2023; Pollard et al. 2018)
Bounded extrapolation	Distribution shift can make predictions unsafe; detect when validation points lie far outside the training envelope	Fraction of validation feature values outside train $[p1, p99]$ per feature within tolerance	(Quionero-Candela et al. 2009)
Conformal coverage (user-specified, here 80%)	Conformal prediction yields finite-sample marginal validity at any target level	Split-conformal intervals target 80% coverage in this study; pass if empirical coverage is 0.80 ± 0.10	(Lei et al. 2018; Angelopoulos and Bates 2023)
Calibration (ECE)	Reliable probabilities are essential for decision-making	ECE ≤ 0.10 (strict) or 0.12 (relaxed) using isotonic/Platt calibration after training	(Guo et al. 2017; Zadrozny and Elkan 2002; Platt 1999)
Temporal consistency	Updated predictions should not behave implausibly over time	After benign improvement (e.g., lactate -0.5), average drop in predicted remaining LOS $\leq$ 12h	(Shickel et al. 2018)
Fairness (group error gap)	Safety also requires equity across subgroups	MAE gap across MICU/SICU/OTHER $\leq \max(6\text{h}, 0.5\times\text{overall MAE})$	(Obermeyer et al. 2019)
Alert-budget band	Excessive alerts cause fatigue and overrides; thresholds must align with staffing capacity	With chosen operating threshold θ, alert rate constrained to 10–40% on validation	(van der Sijs et al. 2006; Ancker et al. 2017)
Occupancy plausibility	Avoid unrealistic census forecasts that would mislead planners	Mean predicted LOS within $k\sigma$ (default k=3) of historical training mean	(operations heuristic; no formal source)

Table 1: Contract collection: rationale, check, and supporting sources. For instance, conformal prediction guarantees coverage at any target level; we set 80% as a conservative operational default.

ranges, guarding against systemic miscalibration distorting bed planning.

How thresholds were chosen. We seeded targets using conservative defaults. For uncertainty, we set 80% coverage as an operational default, noting that conformal prediction can guarantee any chosen level (Lei et al. 2018; Angelopoulos and Bates 2023). For calibration, we required ECE $\leq$0.10–0.12, consistent with isotonic and Platt post-hoc methods (Guo et al. 2017; Zadrozny and Elkan 2002). For deployment, we enforced a 10–40% alert-rate band to mitigate alert fatigue (van der Sijs et al. 2006; Ancker et al. 2017). These thresholds are conservative starting points and can be adapted with stakeholders once data volume and operational constraints are established.

Measures and Statistics

Regression quality is summarized by MAE and coverage of 80% intervals. Binary models report AUC and ECE (binning-based). For fairness, we compute MAE per care unit and the inter-group gap; for operations, we log the realized alert rate and occupancy plausibility. All seeds are fixed; we use patient-level splits to avoid leakage.

Implementation and Reproducibility

All code is pure Python (NumPy/Pandas/Scikit-learn). Contracts are in `icu_contracts_v2.py` and are invoked from the training scripts. Datasets are read from CSVs to avoid database dependencies; when eICU is unavailable and requires a request, a synthetic generator produces `patient.csv` and `lab.csv` with realistic ranges and timestamps. The pipeline provides tables to support exact reproduction of 2–4/5. Runtime on a PC is min per cohort.

Experimental Evaluation

Demo Results using MIMIC-IV

Table 2 summarizes regression performance on the MIMIC-IV demo cohort. The Tweedie GLM baseline achieved a mean absolute error (MAE) of roughly 12 hours, providing a transparent point of comparison but without interval estimates. By contrast, the gradient boosted model (GBM) with conformalized P50 predictions reduced error to about 11 hours and achieved coverage of 0.889 on the $[P_{10}, P_{90}]$ interval, well within the contractual band of 0.80 ± 0.10. Both models passed their specifications, but the conformalized GBM illustrates how uncertainty quantification can be explicitly tied to coverage guarantees, a property that is absent from the GLM.

Table 3 reports classifier calibration results for the binary outcome (*LOS* $\geq$ *48h*). The logistic regression model calibrated with a sigmoid mapping produced excellent calibration error (ECE $= 0.004$) but poor discrimination (AUC ≈ 0.48). In contrast, the GBM showed higher discriminative ability (AUC up to 0.676), yet calibration performance varied: isotonic calibration failed to meet the ECE threshold, while Platt calibration succeeded. Importantly, all models that passed the calibration contract also respected the alert-budget contract, which restricted alerting thresholds to a 10–40% range. The ensemble of the two best-calibrated models balanced discrimination and calibration, illustrating the practical value of combining models to jointly satisfy safety-related requirements. The calibration contract successfully flagged the unsafe GBM–isotonic combination, highlighting how executable specifications expose hidden hazards.

Finally, Table 4 presents results for three system-level contracts. The temporal consistency contract passed, con-

Model	MAE (h)	Coverage [P10,P90]	Contract Pass?
Tweedie GLM	12.3	–	✓
GBM P50 + Conformal	10.8	0.889	✓

Table 2: Regression performance & coverage on the MIMIC-IV demo cohort. Coverage is the fraction of validation stays whose true LOS fell inside the [P10, P90] conformal interval. Contracts enforce target coverage 0.80 ± 0.10.

Model	Calibration	AUC	ECE@5	Theta	AlertRate	ECE Pass	AlertBudget Pass	Contract Pass	ECE Note
LogReg	isotonic	0.489	0.155	0.446	25.9%	✓	✓	✓	relaxed
LogReg	sigmoid	0.481	0.004	0.490	37.0%	✓	✓	✓	strict
GBM	isotonic	0.646	0.143	0.383	37.0%	✗	✓	✗	fail
GBM	sigmoid	0.676	0.053	0.431	33.3%	✓	✓	✓	strict
Ensemble (top-2 by ECE)	avg(prob)	0.602	0.024	0.453	37.0%	✓	✓	✓	strict

Table 3: Classifier calibration & alert budget (MIMIC-IV demo). ECE is computed with 5 bins unless noted as relaxed. Contracts pass only if both calibration and alert-budget conditions are satisfied.

firming that LOS predictions did not increase When the simulated patient's state improved (decreased lactate). The occupancy plausibility contract also passed: the average predicted LOS was within three standard deviations of the historical mean, preventing unrealistic projections for bed occupancy. However, the fairness contract assessed as the gap in MAE across care units failed. MICU predictions were substantially worse than those for SICU or Other units. producing a disparity of nearly 50 hours, far exceeding both the absolute 6-hour bound and the adaptive relative threshold. This contract violation highlights the importance of checking performance in different groups during clinical use. Even if the overall results look good, if there's a lot of error in certain groups, it can lead to safety risks.

Overall, these three tables demonstrate that executable contracts can be used to systematically evaluate not only model fit and calibration, but also fairness, temporal coherence, and operational plausibility. In several cases, contracts flagged configurations that would otherwise appear reasonable from headline metrics alone. This case study, therefore, illustrates how contract-based specifications provide a lightweight but effective mechanism for surfacing hidden hazards in safety-critical AI applications.

Demo Results using eICU

Table 5 summarizes regression performance on the eICU demo cohort. The conformalized gradient boosting model produced a median absolute error of approximately 5.2 hours on the validation set. However, the 80% prediction interval covered only 23.5% of true stays, well below the contractual band of 0.80 ± 0.10. This failure indicates that the conformal radius $\hat{q}$ was too narrow to capture the natural variation in length of stay. In practice, this type of contract violation is critical, since decision makers would be presented with overly confident forecasts that underestimate the true uncertainty.

Table 6 reports classifier calibration results for the binary outcome ($LOS \geq 48h$). Across both logistic regression and gradient boosting models, with isotonic or Platt (sigmoid) calibration, discrimination varied substantially (AUC ranging from 0.046 to 0.785). Calibration error remained modest (ECE < 0.02), but the models consistently failed the alert-budget contract. Alert rates ranged between 40–70%, well outside the specified 10–40% operational band. This highlights the practical value of the alert-budget specification: Although the probabilistic predictions appeared well-calibrated, their thresholded outputs would overwhelm staff with excessive alerts.

Finally, Table 7 evaluates fairness across care units. Mean absolute errors were similar across MICU, SICU, and Other units (5.5, 5.2, and 5.1 hours, respectively), leading to a gap of only 0.42 hours. The fairness contract threshold, computed as the maximum of 6 hours or half of the overall MAE, was satisfied, and the contract passed. This result indicates that, unlike the coverage and alert-budget specifications, subgroup disparities were not a major issue in this demo cohort.

Overall, the eICU demo results highlight several important points. First, contracts surface failures that simple metrics can miss: the regression model looked accurate on point estimates, but its coverage guarantee was violated. Second, calibration metrics alone were insufficient for classifiers; the alert-budget contract exposed that outputs were not operationally usable. Third, the fairness contract reassured that care unit disparities were small, demonstrating that the methodology can detect both problems and successes. Together, the three tables illustrate how executable contracts provide systematic, domain-relevant checks that go beyond headline accuracy when evaluating safety-critical AI models.

Contrasting the eICU demo results with the earlier MIMIC-IV analysis highlights both consistencies and divergences in model behavior across datasets. In the MIMIC-IV cohort, regression coverage in Table 2 closely matched the contractual target, demonstrating that conformal intervals could provide reliable uncertainty estimates. By contrast, the eICU regression results in Table 5 failed the same coverage specification, with prediction intervals that were too narrow. This difference likely reflects the greater heterogeneity of the eICU data, which aggregates patients across

Contract	Metric / Threshold	Result	Pass?
Temporal Consistency	Avg drop ≤ 12 h	0.0 h	✓
Fairness (Careunit MAE gap)	Gap $\leq \max(6\text{ h}, 0.5\times$ overall MAE)	gap=49.6 h | overall MAE=47.8 h | MICU=72.9 (n=12), SICU=37.5 (n=13), OTHER=23.3 (n=7) | threshold=23.9 h (base 6.0, rel $0.5\times$MAE=47.8)	✗
Occupancy Plausibility	$\lvert$mean_pred $-$ hist_mean$\rvert \leq 3 \cdot \sigma$	mean_pred=40.9 h, hist_mean=90.0 h, $\sigma = 100.1$ h	✓

Table 4: System-level contracts (MIMIC-IV demo). Pass indicates whether the specification was satisfied on the validation split.

Split	MAE(P50) [h]	Coverage@80%	Conformal radius $\hat{q}$ [h]	Pass (coverage)
Validation	5.19	0.235	1.65	✗

Table 5: eICU demo: regression MAE and 80% prediction interval coverage on validation.

many hospitals, increasing variance in length of stay patterns compared to the single-center MIMIC-IV dataset.

Both datasets revealed tension between probabilistic metrics and operational constraints for classifier calibration. In MIMIC-IV (Table 3), some models passed both calibration and alert-budget contracts, demonstrating feasible operating points. In the eICU cohort, however, every model in Table 6 failed the alert-budget requirement despite low ECE values. This suggests that multi-center variability not only affects uncertainty estimation for regression but also complicates threshold selection for classifiers.

Finally, fairness contracts showed opposite outcomes. The MIMIC-IV cohort exposed substantial disparities in error across care units (Table 4), leading to a contract violation. In the eICU demo, Table 7 indicated very small differences between MICU, SICU, and Other units, and the contract was satisfied. This contrast underscores the value of auditing multiple datasets: where one dataset suggested fairness risks, the other provided reassurance.

Taken together, these cross-dataset comparisons reinforce the need for contract-based evaluation across diverse settings. Contracts that passed in one cohort (e.g., coverage in MIMIC-IV, fairness in eICU) may fail in another, revealing hidden hazards. Systematic application of executable specifications across datasets, therefore, provides a stronger basis for building trust in safety-critical AI models.

Discussion

We report findings by research question in summary: **RQ1** (Do contracts identify safety issues that accuracy-style metrics miss?), **RQ2** (How do results change between single- and multi-center cohorts?), and **RQ3** (Can simple fixes bring models back into compliance?).

RQ1 (Detection)

Regression uncertainty (coverage) vs. point error. On the single-center MIMIC-IV demo, a conformalized regressor delivered competitive MAE *and* met the 80% prediction-interval requirement (Table 2). The same pipeline on eICU looked reasonable by MAE (5.2 h) yet covered only 23.5% of outcomes—far short of the 0.80 ± 0.10 target (Table 5). In other words, accuracy alone would have passed the model; the coverage contract caught a safety-relevant miss.

Calibration vs. operational alerting. For the binary *LOS* $\geq$ *48h* task, several calibrated MIMIC-IV models achieved low ECE and stayed within the operational alert band (Table 3). In eICU, models with similarly low ECE still triggered 40–70% alerts (Table 3), blowing past the alert budget. Probability calibration, by itself, does not guarantee operational feasibility; the alert-budget contract makes that mismatch visible.

Equity across clinical subgroups. Fairness auditing adds another lens. In MIMIC-IV, the care-unit MAE gap exceeded the contract threshold (Table 4), concentrating error in specific units. In the eICU demo, the gap was small (0.42 h) and the contract passed (Table 4). Either way, the fairness contract provides an explicit, quantitative check that global MAE/AUC cannot.

Takeaway: Coverage, alert-budget, and fairness contracts surfaced failures that standard accuracy or calibration summaries would have overlooked, tying model quality to safety and day-to-day operations.

RQ2 (Portability)

Uncertainty under multi-center heterogeneity. Coverage was attainable in the single-center MIMIC-IV demo (Table 2) but failed in eICU (Table 5). Heterogeneity across hospitals widens residuals and, without retuning, leaves conformal intervals too narrow.

Operational feasibility is more brittle in eICU. MIMIC-IV offered threshold settings that satisfied the alert budget (Table 3); eICU did not, despite low ECE (Table 3). When case mix and practice patterns vary, staying within staffing budgets requires finer threshold control.

Fairness patterns can invert. The fairness contract failed on MIMIC-IV but passed on eICU (Table 4). Subgroup risk is dataset-specific, so audits must be rerun whenever the deployment context changes.

Takeaway: Multi-center variability in eICU stresses uncertainty and operational contracts (coverage and alert budget), and fairness patterns need not match those observed in

Model	AUC	ECE@5	θ	Alert rate	Pass
LogReg+isotonic	0.651	0.012	0.020	70.1%	✗
LogReg+sigmoid	0.785	0.014	0.018	40.1%	✗
GBM+isotonic	0.046	0.012	0.018	51.9%	✗
GBM+sigmoid	0.634	0.014	0.020	40.6%	✗

Table 6: eICU demo: classifier calibration (ECE) and alert budget contract on validation.

MICU MAE [h]	SICU MAE [h]	OTHER MAE [h]	Overall MAE [h]	Gap [h]	Threshold [h]	Pass	N_MICU/N_SICU/N_OTHER
5.50	5.18	5.08	5.19	0.42	6.00	Yes	33/59/95

Table 7: eICU demo: MAE by care unit, overall MAE, gap and threshold; pass indicates group-gap contract satisfied.

single-center data. Contract outcomes are not portable; they must be re-validated per cohort.

RQ3 (Remediation)

Conformal tuning for coverage. Both cohorts used split-conformal intervals around a median regressor. MIMIC-IV met coverage out of the box (Table 2); eICU did not (Table 5). The contract points directly to remedies: enlarge the conformal radius or adopt distribution-aware intervals (e.g., quantile or normalized-residual schemes).

Post-hoc probability calibration and threshold selection. Isotonic or Platt calibration consistently lowered ECE, and quantile-based threshold search helped target feasible alert rates (Table 3). In eICU, calibration alone was insufficient; the alert-budget contract suggested raising the operating threshold or narrowing alert scope to remain within the 10–40% band.

Fairness monitoring with adaptive thresholds. The care-unit gap contract scales its bound with overall error (maximum of 6 h or $0.5\times$MAE), which is lenient for early models yet still informative. Exceeding this bound in MIMIC-IV (Table 4) points to subgroup-specific features or calibration drift; passing it in eICU (Table 4) supports equitable performance under the current setting.

Takeaway: Split-conformal adjustments, post-hoc calibration, and principled thresholding are simple levers that move models toward compliance. When there are not enough (e.g., eICU coverage or alerting), the contracts expose the failure mode and indicate concrete next steps, interval widening, subgroup-aware modeling, or alert-policy redesign.

Summary

Across two cohorts and six tables, the contract suite did what it is meant to do: safety measures and operational expectations into executable checks. It separated look-alike models by their real-world suitability, showed where tuning sufficed, and flagged the situations that require deeper modeling or policy change. Reproducibility materials, including code, trained models, evaluation outputs, and dataset links, are provided in (med 2025).

Conclusion & Future Work

We presented an executable set of safety contracts for clinical prediction models and applied them to hospital length-of-stay. The contracts translate familiar design-by-contract ideas into checks that live alongside the pipeline—covering data integrity and timing, label handling and censoring, uncertainty and calibration targets, operational alert budgets, and subgroup equity. Evaluated on a single-center (MIMIC-IV) and a multi-center (eICU-style) cohort with simple baselines (logistic regression and gradient boosting) plus conformal prediction and post-hoc calibration, the approach consistently surfaced problems that headline metrics conceal. Typical examples include acceptable MAE paired with severe under-coverage, and well-calibrated probabilities that nonetheless produce unsustainable alert rates. In many cases, lightweight remedies tuning conformal radii, adjusting thresholds/alert scope, or refining calibration restored compliance without harming point accuracy; when they did not, the contracts made the failure mode explicit and pointed to concrete next steps. Three key lessons emerge: First, MAE/AUC/ECE alone do not determine a model's usability; contracts make expectations testable. Second, results vary by context, so checks need to be repeated when data distribution or practice patterns change. Third, since each check is inexpensive and local, contracts can function as acceptance tests in development during deployment.

Limitations. This case study uses limited models and data, with literature-based thresholds needing validation and potential site-specific adjustments. It doesn't cover retraining or large-scale continuous monitoring and requires access to substantial representative data.

Future work. We plan to extend our approach for tasks like sepsis and readmission, gather thresholds with clinical stakeholders, explore highly parameterized models like DNN, and enhance temporal contracts and invariants. These efforts shift from "detect and report" to "detect, explain, and act" for better clinical ML safety. Future work can include input from ICU leadership and stakeholders to ensure these contracts align with clinical priorities and operations.

Acknowledgments

This work was supported in part by a startup research fund from Texas State University. We thank the reviewers for their constructive feedback. The OpenAI LLM model was used for the draft revisions, along with LaTeX cleanup, finding references while collecting contracts, and small utilities in the experiment.

References

2025. MedContract Repository. https://github.com/shibbirtanvin/MedContract. [Online; accessed Aug-2025].

Ahmed, S.; Imtiaz, S. M.; Khairunnesa, S. S.; Cruz, B. D.; and Rajan, H. 2023. Design by Contract for Deep Learning APIs. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2023, 94–106. New York, NY, USA: Association for Computing Machinery. ISBN 9798400703270.

Ancker, J. S.; Edwards, A.; Nosal, S.; Bachrach, R.; York, P.; and Kaushal, R. 2017. Effects of Workload, Work Complexity, and Repeated Alerts on Alert Fatigue in a Clinical Decision Support System. *BMC Medical Informatics and Decision Making*, 17(1): 36.

Angelopoulos, A. N.; and Bates, S. 2023. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. *arXiv preprint arXiv:2107.07511*.

Cao, J.; Li, M.; Chen, X.; Wen, M.; Tian, Y.; Wu, B.; and Cheung, S.-C. 2022. DeepFD: Automated Fault Diagnosis and Localization for Deep Learning Programs. In *Proceedings of the 44th International Conference on Software Engineering*, ICSE '22, 573–585. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392211.

Chen, Z.; Cao, Y.; Liu, Y.; Wang, H.; Xie, T.; and Liu, X. 2020. A Comprehensive Study on Challenges in Deploying Deep Learning Based Software. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, 750–762. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370431.

Dolby, J.; Shinnar, A.; Allain, A.; and Reinen, J. 2018. Ariadne: Analysis for Machine Learning Programs. In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, MAPL 2018, 1–10. New York, NY, USA: Association for Computing Machinery. ISBN 9781450358347.

Friedman, J. H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29(5): 1189–1232.

Graham, B.; Furr, W.; Kuczmarski, K.; Biskup, B.; and Palay, A. 2010. PyContracts.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In *Proc. ICML*, 1321–1330.

Humbatova, N.; Jahangirova, G.; Bavota, G.; Riccio, V.; Stocco, A.; and Tonella, P. 2020. Taxonomy of Real Faults in Deep Learning Systems. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, 1110–1121. New York, NY, USA: Association for Computing Machinery. ISBN 9781450371216.

Islam, M. J.; Nguyen, G.; Pan, R.; and Rajan, H. 2019. A Comprehensive Study on Deep Learning Bug Characteristics. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2019, 510–520. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355728.

Islam, M. J.; Pan, R.; Nguyen, G.; and Rajan, H. 2020. Repairing Deep Neural Networks: Fix Patterns and Challenges. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, 1135–1146. New York, NY, USA: Association for Computing Machinery. ISBN 9781450371216.

Johnson, A.; Bulgarelli, L.; Shen, L.; et al. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10(1): 1–21.

Kapoor, S.; and Narayanan, A. 2023. Leakage and the Reproducibility Crisis in Machine-Learning-Based Science. *Patterns*, 4(9): 100791.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Khairunnesa, S. S.; Ahmed, S.; Imtiaz, S. M.; Rajan, H.; and Leavens, G. T. 2023. What Kinds of Contracts Do ML APIs Need? *Empirical Software Engineering*, 1(1).

Lagouvardos, S.; Dolby, J.; Grech, N.; Antoniadis, A.; and Smaragdakis, Y. 2020. Static Analysis of Shape in TensorFlow Programs. In Hirschfeld, R.; and Pape, T., eds., *34th European Conference on Object-Oriented Programming, ECOOP 2020, November 15-17, 2020, Berlin, Germany (Virtual Conference)*, volume 166 of *LIPIcs*, 15:1–15:29. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Lei, J.; G'Sell, M.; Rinaldo, A.; Tibshirani, R. J.; and Wasserman, L. 2018. Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*, 113(523): 1094–1111.

Liu, C.; Lu, J.; Li, G.; Yuan, T.; Li, L.; Tan, F.; Yang, J.; You, L.; and Xue, J. 2021. Detecting TensorFlow Program Bugs in Real-World Industrial Environment. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 55–66.

Lorena Buciu, N. F., Simon Chen; and et al. 2016. PyTA.

McCullagh, P.; and Nelder, J. A. 1989. *Generalized Linear Models*. Chapman and Hall/CRC, 2nd edition.

Mens, T.; Decan, A.; and Spanoudakis, N. I. 2019. A method for testing and validating executable statechart models. *Software & Systems Modeling*, 18(2): 837–863.

Meyer, B. 1992. Applying "Design by Contract". *Computer*, 25(10): 40–51.

Nikanjam, A.; Braiek, H. B.; Morovati, M. M.; and Khomh, F. 2021. Automatic Fault Detection for Deep Learning Programs Using Graph Transformations. *ACM Trans. Softw. Eng. Methodol.*, 31(1).

Obermeyer, Z.; Powers, B.; Vogeli, C.; and Mullainathan, S. 2019. Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations. *Science*, 366(6464): 447–453.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Platt, J. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*. MIT Press.

Pollard, T.; Johnson, A.; Raffa, J.; et al. 2018. The eICU Collaborative Research Database, a freely available multicenter database for critical care research. *Scientific Data*, 5: 180178.

PyCQA. 2016. Pylint: Static Code Analysis for Python. Software documentation.

Quionero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; and Lawrence, N. D. 2009. *Dataset Shift in Machine Learning*. MIT Press.

Riccio, V.; Jahangirova, G.; Stocco, A.; Humbatova, N.; Weiss, M.; and Tonella, P. 2020. Testing machine learning based systems: a systematic mapping. *Empirical Software Engineering*, 25(6): 5193–5254.

Seshia, S. A.; Desai, A.; Dreossi, T.; Fremont, D. J.; Ghosh, S.; Kim, E.; Shivakumar, S.; Vazquez-Chanlatte, M.; and Yue, X. 2018. Formal Specification for Deep Neural Networks. In *Automated Technology for Verification and Analysis*, 20–34. Cham: Springer International Publishing. ISBN 978-3-030-01090-4.

Shickel, B.; Tighe, P.; Bihorac, A.; and Rashidi, P. 2018. Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5): 1589–1604.

van der Sijs, H.; Aarts, J.; Vulto, A.; and Berg, M. 2006. Overriding of Drug Safety Alerts in CPOE. *Journal of the American Medical Informatics Association*, 13(2): 138–147.

Wan, C.; Liu, S.; Hoffmann, H.; Maire, M.; and Lu, S. 2021. Are Machine Learning Cloud APIs Used Correctly? In *Proceedings of the 43rd International Conference on Software Engineering*, ICSE '21, 125–137. IEEE Press. ISBN 9781450390859.

Wardat, M.; Cruz, B. D.; Le, W.; and Rajan, H. 2022. DeepDiagnosis: Automatically Diagnosing Faults and Recommending Actionable Fixes in Deep Learning Programs. In *Proceedings of the 44th International Conference on Software Engineering*, ICSE '22, 561–572. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392211.

Wardat, M.; Le, W.; and Rajan, H. 2021. DeepLocalize: Fault Localization for Deep Neural Networks. In *Proceedings of the 43rd International Conference on Software Engineering*, ICSE '21, 251–262. IEEE Press. ISBN 9781450390859.

Zadrozny, B.; and Elkan, C. 2002. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *Proc. ICML*, 694–701.

Zhang, R.; Xiao, W.; Zhang, H.; Liu, Y.; Lin, H.; and Yang, M. 2020. An Empirical Study on Program Failures of Deep Learning Jobs. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, 1159–1170. New York, NY, USA: Association for Computing Machinery. ISBN 9781450371216.

Zhang, X.; Zhai, J.; Ma, S.; and Shen, C. 2021. AutoTrainer: An Automatic DNN Training Problem Detection and Repair System. In *Proceedings of the 43rd International Conference on Software Engineering*, ICSE '21, 359–371. IEEE Press. ISBN 9781450390859.

Zhang, Y.; Chen, Y.; Cheung, S.-C.; Xiong, Y.; and Zhang, L. 2018. An Empirical Study on TensorFlow Program Bugs. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2018, 129–140. New York, NY, USA: Association for Computing Machinery. ISBN 9781450356992.