

DEPLOYR: A technical framework for deploying custom real-time machine learning models into the electronic medical record

Conor K. Corbin^{*,1}, Rob Maclay^{*,2}, Aakash Acharya³, Sreedevi Mony³, Soumya Punnathanam³, Rahul Thapa³, Nikesh Kotecha, PhD³, Nigam H. Shah, MBBS, PhD⁴, Jonathan H. Chen, MD, PhD⁴

¹Department of Biomedical Data Science, Stanford, California, USA

²Stanford Children's Health, Palo Alto, California, USA

³Stanford Health Care, Palo Alto, California, USA

⁴Center for Biomedical Informatics Research, Stanford, California, USA

* denotes equal contribution

Machine learning (ML) applications in healthcare are extensively researched, but successful translations to the bedside are scant. Healthcare institutions are establishing frameworks to govern and promote the implementation of accurate, actionable and reliable models that integrate with clinical workflow. Such governance frameworks require an accompanying technical framework to deploy models in a resource efficient manner. Here we present DEPLOYR, a technical framework for enabling real-time deployment and monitoring of researcher created clinical ML models into a widely used electronic medical record (EMR) system. We discuss core functionality and design decisions, including mechanisms to trigger inference based on actions within EMR software, modules that collect real-time data to make inferences, mechanisms that close-the-loop by displaying inferences back to end-users within their workflow, monitoring modules that track performance of deployed models over time, silent deployment capabilities, and mechanisms to prospectively evaluate a deployed model's impact. We demonstrate the use of DEPLOYR by silently deploying and prospectively evaluating twelve ML models triggered by clinician button-clicks in Stanford Health Care's production instance of Epic. Our study highlights the need and feasibility for such silent deployment, because prospectively measured performance varies from retrospective estimates. By describing DEPLOYR, we aim to inform ML deployment best practices and help bridge the model implementation gap.

Introduction

Access to real-world data streams like electronic medical records (EMRs) has accelerated the promise of machine learning (ML) applications in healthcare. There exist over 250,000 peer-reviewed articles related to risk-stratification models alone, many of which have been published in the past ten years^{1,2}. Despite the hype, a sizeable gap separates ML models in peer-reviewed research articles from those actually impacting clinical care^{3,4}. This implementation gap leaves most published clinical ML applications lost in the “model graveyard”⁵.

The gap between research and implementation demonstrates that strong predictive performance alone is not sufficient for feasible and worthwhile translation of a clinical ML model to the bedside. Beyond demonstrating predictive accuracy, model champions must articulate concrete actions that can be taken as a result of inferences (model outputs)⁶. Proposed actions must incur utility, some measurably positive effect on a clinical outcome of interest benefiting the deployment population as a whole^{7–9}. Positive impact on average, however, does not imply positive impact for all. Successful ML applications to healthcare must satisfy fairness principles, ensuring performance and accrued utility is not unfavorable across certain patient populations, particularly those traditionally under-served^{10–12}. Institutions leading the charge in the translation of clinical ML applications are establishing governance frameworks to ensure models are safe, reliable and useful throughout their deployment life-cycles^{13–16}.

Even if the above are addressed and a go decision is made, ML applications must overcome technical feasibility hurdles related to their deployment. Traditionally, if such technical implementation were possible at all within the context of existing vendor and legacy infrastructure, massive overhead was required to establish and maintain computational frameworks enabling the deployment of ML models into clinical workflow^{17–19}. Some early adopting institutions rely on frameworks native to their EMR vendor (e.g Epic Cognitive Computing) to deploy ML applications²⁰. These frameworks can reduce technical overhead and facilitate model sharing across health systems, but deployment of custom (researcher developed) ML models trained using an institution's clinical data-warehouse remains difficult. Health

institutions that have successfully developed custom deployment frameworks have largely done so through tight collaborations of data scientists who build and validate models and hospital information technology (IT) teams who know the ins-and-outs of their institution’s EMR²¹. These combinations of expertise are valuable but rare to find. Successful implementations allow flexibility out-of-the-box vendor solutions do not, but design details largely remain internal knowledge, leaving each health system to either re-invent the wheel or be left behind. This has led to a sparse and eclectic set of solutions with limited academic discourse of best practices.

Through a collaboration of data scientists at the Stanford School of Medicine, and IT persona at both Stanford Health Care and Stanford Children’s Health, our objective was to develop DEPLOYR — a technical framework for deploying researcher developed clinical ML applications directly into our EMR’s production environment. In this article, we detail the concrete mechanisms DEPLOYR enables and the design decisions made during its formation to provide a blueprint for development and best practices for execution. The key functions reviewed here include real-time closed-loop model trigger, data retrieval, inference, user interface integration, and continuous monitoring. We then demonstrate the capability and importance of prospective evaluations that extend beyond traditional retrospective analyses by silently deploying twelve ML models triggered by clinician button-clicks in Stanford Health Care’s production instance of Epic.

Methods

Effective deployments of clinical ML models require implementation of several core mechanisms. Here we detail these functions and discuss design decisions considered in our framework’s development. Core functions include data sourcing (at train and inference time), mechanisms that trigger model inference in production, and mechanisms that integrate inferences and recommendations back to end-users. We detail our implementation of a monitoring module that tracks performance of deployed models over time, mechanisms that enable silent trial deployments, and prospective evaluation of model impact. Our deployment framework (DEPLOYR) enables these mechanisms by leveraging capabilities native to Stanford Health Care’s EMR (Epic Systems) in conjunction with three distinct software applications: *DEPLOYR-dev* (a python package used for model development and validation), *DEPLOYR-serve* (a python Azure Function application used to expose trained models as APIs), and *DEPLOYR-dash* (a dashboard implemented using the streamlit python package)^{22,23}. Figure 1 provides a system level overview of the steps involved in a DEPLOYR enabled deployment.

Data sources

Any ML deployment framework requires decisions on how to source data at both train and inference time. To effectively translate research models, the framework should expect training data to be sourced from research grade clinical data-warehouses. At inference time, many ML use cases require access to real-time data streams^{16,24–26}. Since clinical data warehouses are typically several transformations removed from real-time data streams, data mappings must be implemented between the two sources to ensure models receive the same data elements in training and deployment environments.

Training data source DEPLOYR uses data from Stanford’s clinical data warehouse (STARR) for training²⁷. STARR contains de-identified EMR data from over 2.4 million unique patients spanning 2009-2021 who have visited Stanford Hospital (academic medical center in Palo Alto, California), ValleyCare hospital (community hospital in Pleasanton, California) and Stanford University Healthcare Alliance affiliated ambulatory clinics. STARR tables are hosted on Google BigQuery through a secured Google Cloud instance safe for protected health information (PHI) and maintained by the Stanford School of Medicine²⁸. STARR data relevant for model construction include patient demographics, comorbidities, procedures, medications, laboratory results, and vital signs. Use of STARR data was approved by the institutional review board of the Stanford University School of Medicine.

Inference data source In production, we source data directly from the EMR’s transactional database, Epic Chronicles, which contains real-time patient data²⁹. We access Epic Chronicles using Epic and FHIR (Fast Healthcare

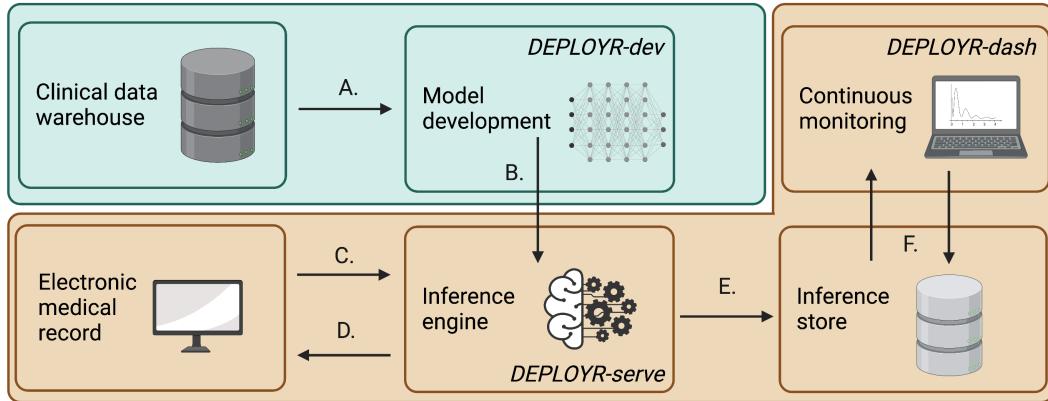


Figure 1: Summary of a DEPLOYR enabled model deployment. Blue shading indicates infrastructure operated by the Stanford School of Medicine (academic research). Orange shading indicates infrastructure operated by Stanford Health Care (clinical operations). A. De-identified EMR data is sourced from Stanford’s clinical data warehouse (STARR), a model is developed and retrospectively validated using the *DEPLOYR-dev* python package. B. The model is deployed to the inference engine, and exposed as a REST API using *DEPLOYR-serve*. C. Inference is triggered, spawning an HTTPS request from the EMR directed at the exposed model. D. The request results in the collection of a feature vector from the EMR’s transactional database using REST (both FHIR and EMR specific) APIs maintained by the EMR vendor. Inference is performed on the real-time retrieved data, and routed back to the EMR closing the loop with end-users and integrating into workflow. E. Inferences and relevant metadata are additionally saved to the inference store, (F.) and consumed by monitoring software (*DEPLOYR-dash*) that continuously tracks model performance via dashboard. REST = Representation State Transfer. HTTPS = Hypertext Transfer Protocol Secure. FHIR = Fast Healthcare Interoperability Resources.

Interoperability Resources) REST APIs documented in Epic’s App Orchard.^{30,31} Credentials enabling authentication to these APIs were provisioned upon registration of a back-end application to Epic’s App Orchard. HTTPS requests directed at these APIs enable real-time access to commonly used clinical ML features, including patient demographics, comorbidities, procedures, medications, laboratory results and vital signs. Use of this data source falls under the umbrella of hospital operational work that is specifically scoped to quality improvement. Data sources and mappings are depicted in Figure 2.

Model inference triggers

Deployment frameworks require mechanisms to trigger model inference. The choice of a suitable inference triggering mechanism depends on how models expect to integrate with clinical workflow. Because triggering logic specifies a model’s deployment population (determines which patients under which conditions receive an inference), it should be considered during cohort development to ensure the population in a researcher’s retrospective test set matches what is seen in production. DEPLOYR supports two broad classes of inference triggering mechanisms — event and time based triggers.

Event based triggers Event based triggers execute as a direct result of a clinical action. Examples include order entry for a laboratory diagnostic test, signature of a progress note, inpatient admission, or discharge. We enable event based triggers by exposing deployed models as REST APIs that listen for inbound HTTPS requests from Epic. Models are wrapped in custom python functions exposed as REST APIs using *DEPLOYR-serve*, an Azure Function application deployed to Stanford Health Care’s instance of Azure²². HTTPS requests are spawned from the EMR through use of Epic Best Practice Advisories (BPAs), rules, and programming points configured to execute upon button clicks in Epic Hyperspace, Epic’s graphical user-interface^{32,33}. HTTPS requests from Epic transmit patient identifiers to *DEPLOYR-serve* functions enabling feature vector collection specific to the patients for whom inference

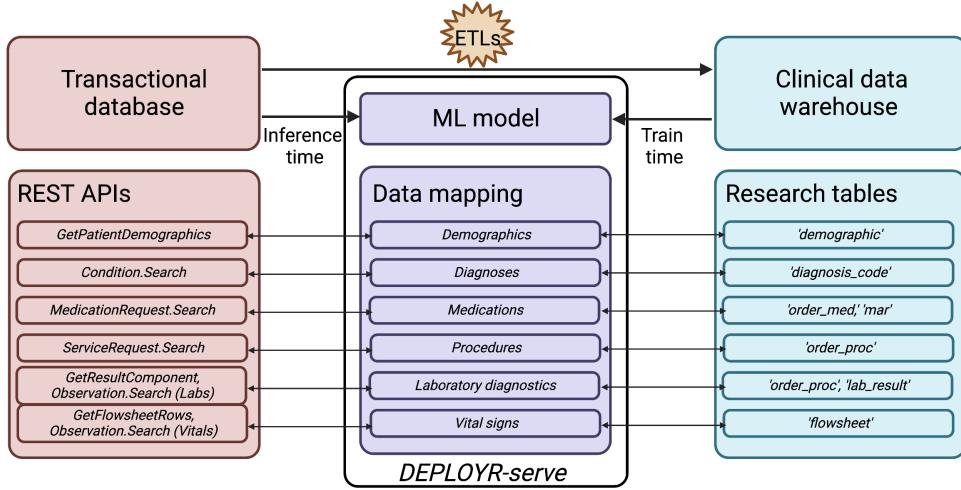


Figure 2: ML models are trained using data sourced from the Stanford’s clinical data warehouse (STARR). In production, real-time data is sourced from the EMR’s transactional database (Epic Chronicles) through Epic and FHIR REST APIs. STARR data are several ETLs (extract, transform, loads) removed from the transactional database. Data mapping is necessary at inference time to ensure features seen during training match features seen in production. Mappings and inferences are invoked in *DEPLOYR-serve*.

was triggered. *DEPLOYR-serve* functions are exposed only to traffic within Stanford Health Care’s internal network.

Time based triggers Time based triggers initiate inference at preset time intervals on batches of patients at once. Example applications could include models designed to monitor patients for signs of deterioration, sepsis, or acute kidney injury by performing inference on collections of patients every fifteen minutes.^{34–37}. *DEPLOYR-serve* supports time based triggering through the use of Azure Function timer triggers, which we configure using cron logic^{22,38}. Unlike event based triggers that execute inference for specific patients, time based triggers initiate inference on batches. Because time based triggers do not originate from Epic, ML models that use them must first identify for which patients inference should be performed. *DEPLOYR-serve* functions fetch this information using Epic APIs that collect batches of patients identifiers, for example, within specific hospital units. Event and time based triggers are summarized in Figure 3.

Directing model outputs

Deployment frameworks require mechanisms to direct model inferences and recommendations back to end-users, closing the loop. Although it is possible to communicate inferences via third-party web or mobile applications, We integrate inferences directly into the EMR to reduce technological overhead and best fit with clinical workflow. Inference integration with Epic is possible through several distinct mechanisms. Inference integration is implemented in *DEPLOYR-serve*.

Passive integration Model inferences can be written back to the EMR without interrupting clinical workflow. An example includes directing inferences to Epic displayed as external model score columns visible in inpatient lists and outpatient schedules. Score columns can be configured to display predicted probabilities, binary flags, feature values and their contributions (e.g., Shapley values) to promote inference level interpretability³⁹. Additionally, inferences can be written to flowsheet rows and smart data values, which can be used to trigger downstream clinical decision support⁴⁰. Inferences and suggested interventions can also be directed at clinicians using Epic’s in-basket messaging platform. Inferences are routed back to the EMR by way of HTTPS requests directed at APIs exposed and documented by Epic. Writing to external model score columns requires the Epic Cognitive Computing API²⁰. Writing to smart

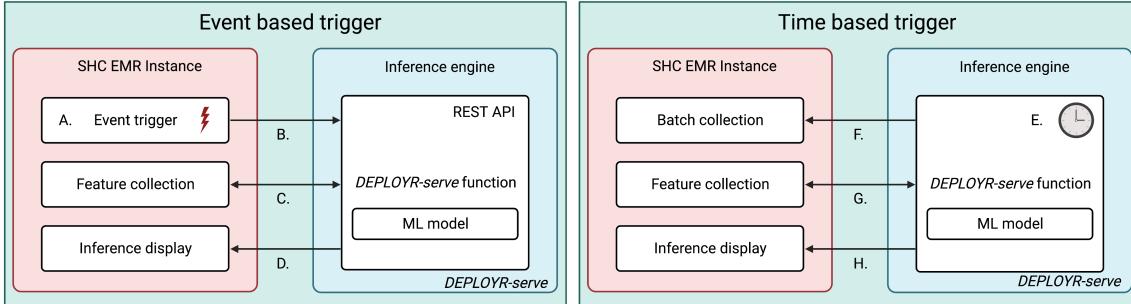


Figure 3: DEPLOYR triggering mechanisms. Models deployed with event based triggering logic are exposed as REST APIs on the inference engine using a python Azure Function application (*DEPLOYR-serve*). An event (A) in the EMR (e.g clinician button-click initiating a laboratory order) transmits an HTTPS request (B) directed at the exposed *DEPLOYR-serve* function, which wraps an ML model. The function transmits HTTPS requests (C) to REST APIs documented in Epic’s App Orchard to collect a feature vector, performs model inference, and directs the inference and resulting clinical decision support via HTTPS request (D) back into the EMR to interface with end-users. Models deployed with time based triggering logic perform inference at set intervals (E) through use of Azure Function timer triggers. Every time interval (e.g 15 minutes), a *DEPLOYR-serve* function transmits HTTPS requests (F) to REST APIs to retrieve a batch of patient identifiers for whom inference should be made. Feature vectors are collected for the batch of patients (G), and inferences are transmitted back into the EMR (H). SHC = Stanford Health Care.

data values, flowsheet rows and sending in-basket messages does not.

Active integration Some applications will better integrate with workflow through interruptive alerts in the clinical work-stream. Such active integration is possible through use of typical EMR alerts. Specifically, we use Epic BPA (Best Practice Advisory) web services. Epic supports two styles of web services: classic CDA (clinical document architecture) and CDS web-hooks⁴¹. DEPLOYR currently uses classic CDA web services, which direct HTTPS requests at *DEPLOYR-serve* functions and await XML (Extensible Markup Language) responses⁴². Data packaged in the XML response is then displayed as an alert in the EMR’s user interface. We show mock-ups of EMR integration capabilities in Figure 4.

Continuous performance monitoring

Once ML models are deployed, they must be monitored to ensure continued reliability in production^{43–45}. All ML models are subject to potential performance decay over time due to changes in the underlying data distribution. Examples including covariate shift (changes in the distribution over features), label shift (changes in distribution over labels) and concept shift (changes in the distribution of labels conditioned on features)^{2,46}. Suitable deployment infrastructure requires monitoring software to track model performance metrics, as well as feature and label distributions throughout the model’s life cycle.

Extracting labels in production To track ML performance in production, software must be implemented to collect labels. DEPLOYR uses *LabelExtractors* to perform this task. *LabelExtractors* are implemented in *DEPLOYR-serve* and are specific to a particular deployed model. They execute at user-defined frequencies using cron logic. At inference time, in addition to directing model outputs to Epic, we package inferences with relevant metadata (identifiers, timestamps, features) in inference packets (JSON objects) and save them to the inference store, an Azure Cosmos database maintained by Stanford Health Care. *LabelExtractors* consume inference packets from the inference store and pair to them their corresponding labels (once observable) using Epic and FHIR APIs. A model tasked with predicting unplanned (30-day readmission) might produce inference packets that include the patient’s FHIR identifier and discharge time (inference time) along with the inference itself. The corresponding *LabelExtractor* would then use the

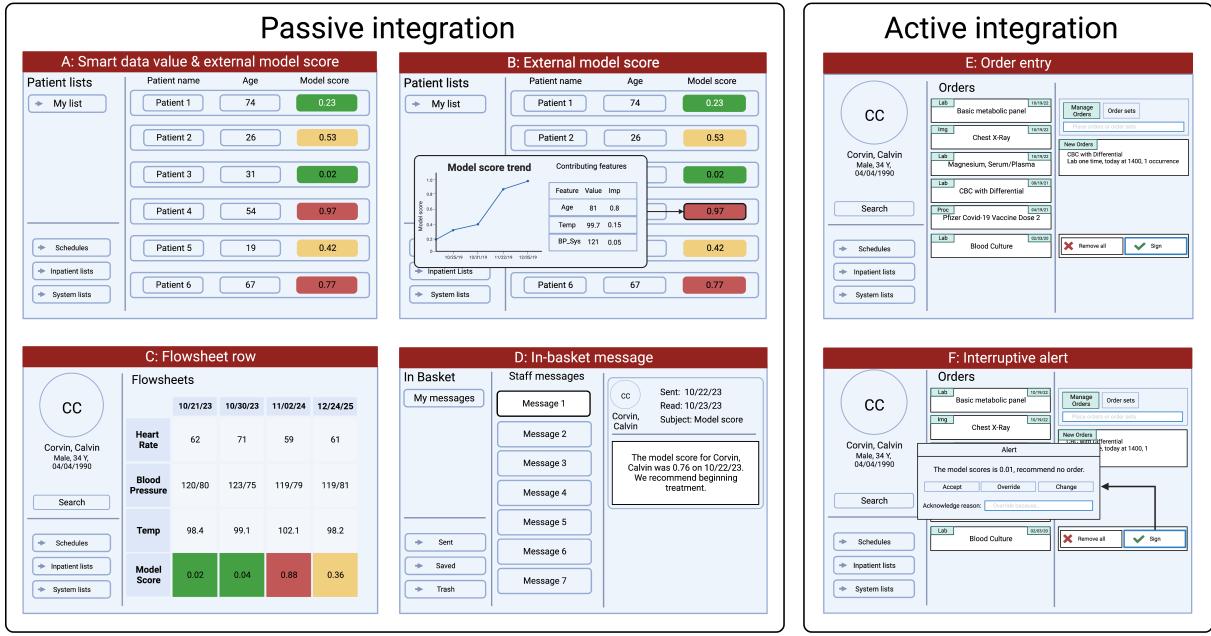


Figure 4: Mock-up frames depicting the EMR user-interface and mechanisms in which inferences (model outputs) can be displayed to end-users. Inferences can be directed back into the EMR passively, without interrupting clinical workflow. They can be written as smart data values or as external model score columns displayed in inpatient lists and outpatient schedules (A). Hovering the mouse over an external model score column value in a patient list displays a pop-up window (B) depicting the trend of the model score over time, feature values, and contribution of those features to the resulting inference. Model inferences can be written to flowsheet rows (C) and visualized over time in conjunction with other vital sign data (e.g. heart rate, blood pressure, temperature). Inferences and suggested interventions can be directed as in-basket messages (D) to specific providers. Additionally, inferences can be integrated with the EMR actively through use of interruptive alerts (E, F) that trigger as a result of button-clicks (e.g. signature of a laboratory order) in the user-interface. Inference integration is implemented in *DEPLOYR-serve*.

FHIR identifier and discharge time to make HTTPS requests directed at relevant APIs, determining whether the patient was readmitted to the hospital within 30 days of the produced inference.

Tracking model performance After a *LabelExtractor* has paired a set of inference packets to their corresponding labels, standard model performance metrics can be estimated with the prospectively collected data. When monitoring binary classifiers, standard performance metrics may include threshold dependent measures like accuracy, sensitivity (recall), specificity, positive predictive value (precision) along with threshold independent metrics like area under the receiver operator characteristics curve (AUROC) and average precision. Standard performance plots that track discrimination ability (ROC and precision-recall curves) and calibration (calibration curves) can be constructed. In addition to full population estimates, these metrics can be continuously tracked over patient subgroups, particularly those belonging to protected demographic classes. Beyond classic ML metrics, measures of model usefulness such as net benefit and expected utility can be tracked to ensure model use is yielding more good than harm.^{7,47}

Tracking distribution shifts in features and labels Machine learning performance decay over-time is often attributable to changes in the underlying data distribution. Beyond calculating prospective performance metrics, DEPLOYR monitoring infrastructure tracks statistics (e.g mean of data elements across groups of patients) that describe the distributions of features, labels and predictions over time. When deviations in model performance arise, corresponding changes to underlying feature and label distributions can be the reason. Both model performance and

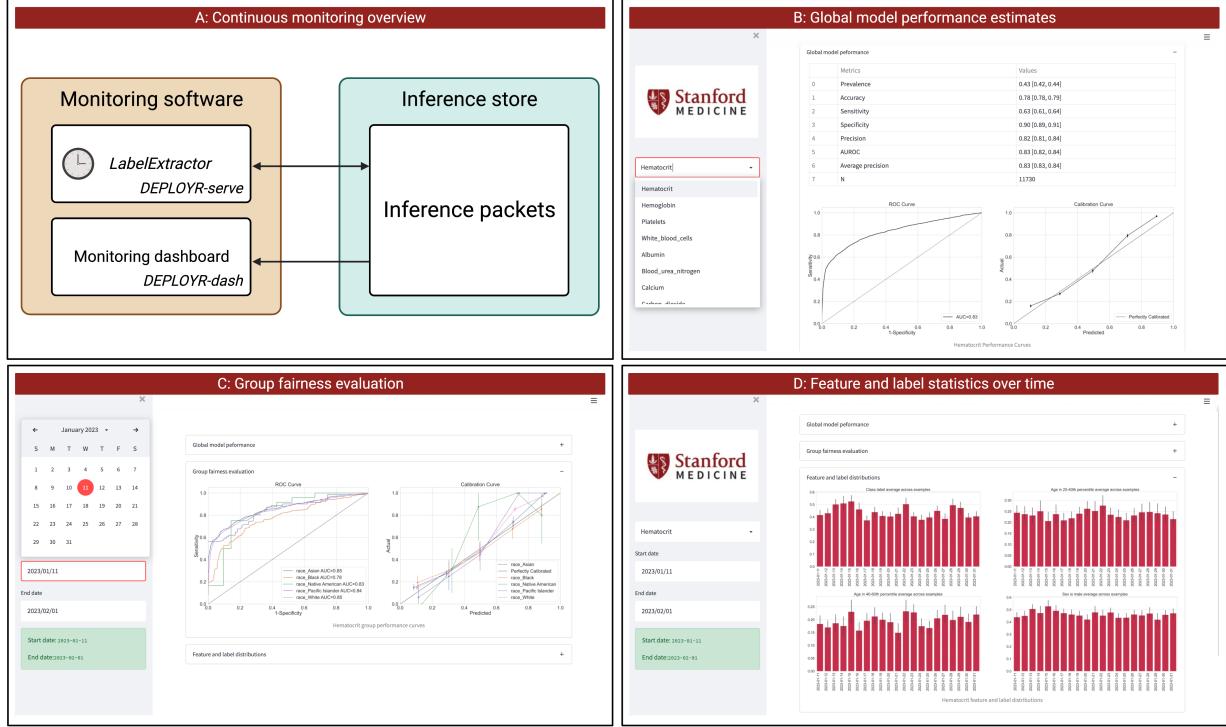


Figure 5: DEPLOYR performance monitoring. Panel A depicts the flow of data from monitoring software to the inference store. A *LabelExtractor* is implemented in *DEPLOYR-serve* as an Azure Function timer trigger. When executed it collects inference packets and pairs them with their corresponding labels. Inference packets paired with labels are consumed by a monitoring dashboard, implemented in *DEPLOYR-dash*. Panel B shows a screenshot of the dashboard displaying global model performance for the user-specified model. Panel C shows a group fairness evaluation across protected demographic classes for the user-specified time window. Panel D shows feature and label distribution statistics collected and tracked over time.

distributional statistics are displayed to a dashboard created using the streamlit python package, which we show in Figure 5²³.

Enabling silent deployment

Suitable deployment infrastructure requires the ability to silently deploy models prospectively to ensure reliability in production before “loud” deployment when inferences are displayed to end-users^{14,48}. Silent trials allow a data science team responsible for model deployment to ensure data pipes have been appropriately linked, especially useful when training and deployment data sources differ. Additionally, silent deployment can uncover faulty cohort design, for instance when a retrospective cohort was generated using exclusion criteria observable only after inference time, common when generating ML cohorts using case-control study design^{49,50}. DEPLOYR event and time based triggers can be configured to execute in the background, enabling silent trials.

Enabling prospective evaluation of impact

The ultimate evaluation of healthcare machine learning impact is not in a prospective ROC curve or simulation of clinical utility, but estimation of the causal effect of the model’s implementation on clinical and operational outcomes. In some use-cases, a model’s inference and corresponding recommendation will only be displayed to the end-user if a predicted risk exceeds some threshold value. In these situations, regression discontinuity designs may be suitable

Table 1: Demographic breakdown of retrospective and prospective cohorts. Retrospective cohorts were sourced from Stanford’s clinical data warehouse (STARR). Prospective cohorts were collected in real-time through Epic Chronicles as diagnostic orders triggered model inferences between January 11 and February 15, 2023. CBC=Complete blood count with differential.

Demographic breakdown		Retrospective cohorts			Prospective cohorts		
Diagnostic		CBC	Magnesium	Metabolic	CBC	Magnesium	Metabolic
N unique patients		13362	11771	13410	18982	5234	16441
Age, mean (SD)		51.4 (23.7)	53.1 (24.2)	54.6 (21.4)	55.7 (21.4)	62.1 (17.8)	55.4 (21.1)
Sex, n (%)	Female	7103 (53.2)	5432 (46.1)	6923 (51.6)	10092 (53.2)	2425 (46.3)	8711 (53.0)
	Male	6259 (46.8)	6338 (53.8)	6485 (48.4)	8884 (46.8)	2808 (53.6)	7724 (47.0)
	Unknown	0 (0.0)	1 (0.0)	2 (0.0)	6 (0.0)	1 (0.0)	6 (0.0)
Race, n (%)	White	6888 (51.5)	6038 (51.3)	6915 (51.6)	9009 (47.5)	2726 (52.1)	7636 (46.4)
	Other	2870 (21.5)	2899 (24.6)	2680 (20.0)	4545 (23.9)	1067 (20.4)	4103 (25.0)
	Asian	2444 (18.3)	1739 (14.8)	2600 (19.4)	3612 (19.0)	928 (17.7)	3125 (19.0)
	Black	557 (4.2)	587 (5.0)	583 (4.3)	1060 (5.6)	327 (6.2)	884 (5.4)
	Unknown	391 (2.9)	238 (2.0)	420 (3.1)	382 (2.0)	56 (1.1)	353 (2.1)
	Pacific Islander	168 (1.3)	212 (1.8)	171 (1.3)	294 (1.5)	98 (1.9)	261 (1.6)
	Native American	44 (0.3)	58 (0.5)	41 (0.3)	80 (0.4)	32 (0.6)	79 (0.5)

to estimate the local treatment effect of the model’s implementation for patients with predicted risk near the threshold⁵¹. In other use-cases when a model’s inference is displayed irrespective of the value it takes, or when an average treatment effect on the deployment population at large is desired, randomized study designs may be required^{52,53}. DEPLOYR supports functionality to inject randomization into the inference-directing mechanism to enable prospective determination of a model’s impact.

Results

We used the DEPLOYR framework to silently deploy twelve binary ML classifiers that predict laboratory diagnostic results, building off of previous retrospective analyses of models designed to reduce wasteful laboratory utilization^{26,54–56}. Models were trained with de-identified EMR data from Stanford’s clinical data warehouse (STARR) using *DEPLOYR-dev*, and exposed as REST APIs using *DEPLOYR-serve*. Epic BPAs were configured to spawn HTTPS requests directed at our deployed models upon order signature of the diagnostic test whose result each model aimed to predict — instances of event-based triggering mechanisms.

Cohort and prediction task definitions

We constructed three retrospective cohorts with STARR data to develop and validate twelve binary ML classifiers. Cohorts were specific to the laboratory diagnostic exam whose results our models predict. The unit of observation was an order for a diagnostic exam. One cohort included orders for complete blood count (CBC) with differential diagnostics. Another included orders of metabolic panels. A third included orders for magnesium diagnostics. CBC and metabolic panel diagnostics result in several components, whereas magnesium tests yield a single result. Consistent with prior work, we trained independent binary classifiers per component to, at order time, predict whether the test result would fall outside the clinical laboratory defined normal reference range^{26,55}. The CBC cohort was used to train four models that separately predicted hematocrit, hemoglobin, white blood cell, and platelet results. The metabolic panel cohort was used to train seven models that predicted albumin, blood urea nitrogen, calcium, carbon dioxide, creatinine, potassium, and sodium results. We trained one model to predict magnesium results. All three retrospective cohorts contained data from 2015 to 2021, with two thousand diagnostic tests sampled randomly per year for a total of 14,000 orders per task. Corresponding prospective cohorts were collected in real-time during our silent deployment trial — spanning the dates January 11 to February 15, 2023. Prospective cohorts were not used to further tune our models, but rather served as additional test sets for model evaluation collected directly in the production environment. Demographic characteristics of patients included in both retrospective and prospective cohorts are summarized in Table 1.

Table 2: Model performance estimates on retrospective and prospectively collected test sets for all twelve models. AUROC=area under the receiver operating characteristics curve.

Model performance		Positive (abnormal) prevalence		AUROC	
Diagnostic	Component	Retrospective	Prospective	Retrospective	Prospective
CBC with differential	Hematocrit	0.47 [0.45, 0.49]	0.43 [0.42, 0.44]	0.86 [0.85, 0.88]	0.83 [0.83, 0.84]
	Hemoglobin	0.50 [0.48, 0.52]	0.47 [0.46, 0.47]	0.88 [0.86, 0.89]	0.83 [0.83, 0.84]
	Platelets	0.26 [0.24, 0.28]	0.21 [0.21, 0.22]	0.79 [0.77, 0.82]	0.77 [0.76, 0.78]
	White blood cell	0.29 [0.27, 0.31]	0.27 [0.27, 0.28]	0.76 [0.74, 0.79]	0.69 [0.68, 0.70]
Metabolic panel	Albumin	0.20 [0.18, 0.21]	0.21 [0.20, 0.21]	0.88 [0.86, 0.91]	0.85 [0.84, 0.86]
	Blood urea nitrogen	0.22 [0.20, 0.24]	0.24 [0.23, 0.25]	0.85 [0.83, 0.87]	0.80 [0.79, 0.81]
	Calcium	0.11 [0.10, 0.13]	0.11 [0.11, 0.12]	0.80 [0.76, 0.83]	0.79 [0.78, 0.81]
	Carbon dioxide	0.16 [0.14, 0.17]	0.20 [0.20, 0.21]	0.69 [0.66, 0.72]	0.62 [0.61, 0.63]
	Creatinine	0.31 [0.29, 0.33]	0.32 [0.31, 0.33]	0.78 [0.75, 0.80]	0.75 [0.74, 0.76]
	Potassium	0.06 [0.05, 0.08]	0.08 [0.08, 0.09]	0.67 [0.61, 0.72]	0.60 [0.59, 0.62]
	Sodium	0.12 [0.10, 0.13]	0.17 [0.16, 0.17]	0.79 [0.75, 0.82]	0.71 [0.70, 0.72]
	Magnesium	0.15 [0.14, 0.17]	0.14 [0.13, 0.15]	0.70 [0.67, 0.73]	0.65 [0.63, 0.67]

Model training and evaluation

Random forest classifiers were fit for each task. Features included patient demographics, diagnosis codes mentioned on the problem list, medication orders, and prior lab results. Features were represented as counts⁵⁷, with numerical data first binned into quintiles based on distributions observed in the training set — as detailed in Supplemental Note 1. All twelve ML models were evaluated using retrospective and prospective test sets. We measured model discrimination ability by estimating the area under the receiver operating characteristics curve (AUROC). 95% confidence intervals were estimated by bootstrapping the corresponding test set 1000 times. We report these estimates along with the prevalence of the positive class (an abnormal diagnostic result) in Table 2. Additionally, we constructed retrospective and prospective receiver operating characteristics (ROC) curves, precision recall (PR) curves, and calibration plots — shown in the Supplemental Figures 6-8. Beyond tracking global performance measures, we estimate model performance across patient subgroups stratified by protected demographic classes in both retrospective and prospective test sets. These results are shown in the Supplemental Tables 3-14.

Discussion

In this study we develop and demonstrate DEPLOYR, a technical framework for moving ML models trained on EMR data from research to production. DEPLOYR was developed through a collaboration of researchers within the Stanford School of Medicine and IT persona at Stanford Health Care and Stanford Children’s Health. We detail design configurations, and evaluate our framework by developing, silently deploying and prospectively evaluating twelve ML models into Stanford Health Care’s production instance of Epic.

To facilitate translation of researcher developed models, data is sourced from Stanford’s clinical data warehouse at train time. At inference time, data is sourced from our EMR’s transactional database. Though possible to perform inference using other sources (e.g analytic or enterprise data warehouses), model inferences would not incorporate up-to-date data. Examples of use cases requiring real-time date are plentiful, ranging from sepsis prediction and early warning scores, to medication and laboratory diagnostic recommender systems, to emergency department triage^{25,26,34-37,55,58-63}.

Related to the need for real-time data is the need for flexible inference triggering mechanisms. Event-based triggers allow inference to spawn from button-clicks within our EMR’s user interface (Epic Hyperspace). Example use cases include ML models that suggest diagnostic tests upon a patient admission, suggest minimizing wasteful diagnostics upon order entry, and suggest diagnosis coding upon note signatures^{26,64-69}. In other ML applications, inference is better integrated into workflow when executed on batches of patients — for example a sepsis prediction model that monitors all patients in the ICU every fifteen minutes⁷⁰. We enable these workflows through time based triggers. To close the loop, we integrate inferences directly into the EMR, eliminating the need for external user facing applications

that can increase friction and slow adoption.

Continuous monitoring tools are essential to any deployment framework⁴³. *LabelExtractors* pair inferences with their corresponding labels when observable, and inference-label pairs are consumed by software that estimate relevant ML performance measures and statistics that describe the distribution of features, labels and predictions over time. Statistically significant and clinically meaningful variations in performance can serve as indicators suggesting models be re-trained or decommissioned⁷¹⁻⁷³. When monitoring deployed ML applications, careful attention to feedback mechanisms from interventions administered as a result of model integration must be considered. Feedback mechanisms when ignored can cause drastic deviations between observed and actual performance⁷⁴⁻⁷⁶. Naively updating these models do more harm than good⁷⁷. DEPLOYR supports injecting randomization into the inference routing mechanism, which can be used in tandem with traditional weighting estimators to recover true performance estimates in the presence of feedback⁷⁸. The ability to randomize is similarly critical for deliberate randomized controlled trial evaluations of ML model impact on clinical outcomes^{52,53}.

Silent trials are necessary before ML applications are integrated into clinical work-streams¹⁴. Deployment frameworks require mechanisms to trigger inference using production data without interacting with clinical work streams. As demonstrated by our silent trial, ML performance estimated on clinical data warehouse derived retrospective test sets can vary from prospectively observed performance in production. Across our twelve laboratory predictions tasks, AUROC in our prospective test sets were generally several percentage points lower than what is seen retrospectively. Poorer performance could be attributable to data drift, data-elements appearing to be accessible at inference time in the clinical data-warehouse not actually being accessible in production, and imperfect mappings between training and inference data sources. Due to these deviations, we recommend using performance measures estimated from prospectively collected test sets during silent trials to make final go decisions when deploying clinical ML models.

Limitations to this study include that, while the DEPLOYR framework allows rapid deployment of models, it assumes model training will occur using data sourced from Stanford's common data model. Additionally, DEPLOYR is currently configured to integrate specifically with EMR software provided by Epic Systems and would require additional API mappings and wrappers to transfer to other institutions using different EMR systems and clinical data warehouse structures. Nevertheless, FHIR APIs were used when possible, which due to the 21st Century Cures Act have mandated support across U.S health institutions⁷⁹. Further, the design decisions and considerations we have outlined in this study remain relevant to any institution attempting to install a deployment framework of their own. DEPLOYR currently only supports deployment of ML applications trained using EMR data. While this can even include unstructured information represented in clinical notes, future upgrades would be necessary to support a broader array of clinical ML applications that use multiple data modalities⁸⁰⁻⁸².

Conclusion

There exist frameworks to govern and promote the implementation of accurate, actionable and reliable models that integrate with clinical workflow^{14,15}. The creation of such frameworks has created the need for accompanying technical solutions that enable executing best practices laid out as desiderata⁸³. To enable adherence to such guidance, we have developed and demonstrated DEPLOYR, a technical framework for rapidly deploying as well as prospectively monitoring custom, researcher developed ML models trained using clinical data warehouses. The framework's design elements and capabilities are demonstrated via silent deployment of multiple ML models triggered by button-clicks in a production instance of Epic, enabling prospective evaluation of the models' performance on unseen data directly in the deployment environment. The ability to perform randomized prospective evaluations of researcher created ML models supports the adoption of best practices that can close the ML implementation gap.

Acknowledgements and funding We used data provided by STARR, “STAnford medicine Research data Repository”, a clinical data warehouse containing de-identified Epic data from Stanford Health Care (SHC), the University Healthcare Alliance (UHA) and Packard Children’s Health Alliance (PCHA) clinics and other auxiliary data from Hospital applications such as radiology PACS. The STARR platform is developed and operated by the Stanford Medicine Research IT team and is made possible by Stanford School of Medicine Research Office. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH or Stanford Health Care.

Figures 1-5 were created with BioRender.com. This work was funded in part from the NIH/National Institute on Drug Abuse Clinical Trials Network (UG1DA015815–CTN-0136), Stanford Artificial Intelligence in Medicine and Imaging—Human-Centered Artificial Intelligence Partnership Grant, Doris Duke Charitable Foundation - Covid-19 Fund to Retain Clinical Scientists (20211260), Google Inc (in a research collaboration to leverage health data to predict clinical outcomes), and the American Heart Association - Strategically Focused Research Network - Diversity in Clinical Trials.

Conflict of interests JHC reported receiving consulting fees from Sutton Pierce and Younker Hyde MacFarlane PLLC and being a co-founder of Reaction Explorer LLC, a company that develops and licenses organic chemistry education software using rule-based artificial intelligence technology. CKC reported receiving consultation fees from Fountain Therapeutics.

Code availability The DEPLOYR framework uses three distinct software repositories, *DEPLOYR-dev* to train and validate models, *DEPLOYR-serve* to expose models to the inference engine and *DEPLOYR-dash* to monitor and visualize prospective model performance characteristics. While this software is specifically designed to interact with Stanford data models and systems, we make *DEPLOYR-dev*, *DEPLOYR-dash* and an example of *DEPLOYR-serve* publicly available for reference.

- *DEPLOYR-dev*: <https://github.com/HealthRex/deployr-dev>
- *DEPLOYR-serve*: <https://github.com/HealthRex/deployr-serve>
- *DEPLOYR-dash*: <https://github.com/HealthRex/deployr-dash>

References

- [1] Challener DW, Prokop LJ, Abu-Saleh O. The proliferation of reports on clinical scoring systems: issues about uptake and clinical utility. *Jama*. 2019;321(24):2405-6.
- [2] Guo LL, Pfohl SR, Fries J, Posada J, Fleming SL, Aftandilian C, et al. Systematic review of approaches to preserve machine learning performance in the presence of temporal dataset shift in clinical medicine. *Applied clinical informatics*. 2021;12(04):808-15.
- [3] Chen JH, Asch SM. Machine learning and prediction in medicine—beyond the peak of inflated expectations. *The New England journal of medicine*. 2017;376(26):2507.
- [4] Matheny M, Israni ST, Ahmed M, Whicher D. Artificial intelligence in health care: The hope, the hype, the promise, the peril. Washington, DC: National Academy of Medicine. 2019.
- [5] Seneviratne MG, Shah NH, Chu L. Bridging the implementation gap of machine learning in healthcare. *BMJ Innovations*. 2020;6(2).
- [6] Callahan A, Shah NH. Machine learning in healthcare. In: Key Advances in Clinical Informatics. Elsevier; 2017. p. 279-91.
- [7] Shah NH, Milstein A, Bagley SC. Making machine learning models clinically useful. *Jama*. 2019;322(14):1351-2.
- [8] Ko M, Chen E, Agrawal A, Rajpurkar P, Avati A, Ng A, et al. Improving hospital readmission prediction using individualized utility analysis. *Journal of Biomedical Informatics*. 2021;119:103826.
- [9] Jung K, Kashyap S, Avati A, Harman S, Shaw H, Li R, et al. A framework for making predictive models useful in practice. *Journal of the American Medical Informatics Association*. 2021;28(6):1149-58.
- [10] Hernandez-Boussard T, Bozkurt S, Ioannidis JP, Shah NH. MINIMAR (MINimum Information for Medical AI Reporting): developing reporting standards for artificial intelligence in health care. *Journal of the American Medical Informatics Association*. 2020;27(12):2011-5.
- [11] Lu JH, Sattler A, Wang S, Khaki AR, Callahan A, Fleming SL, et al. Considerations in the reliability and fairness audits of predictive models for advance care planning. *medRxiv*. 2022.
- [12] Char DS, Shah NH, Magnus D. Implementing machine learning in health care—addressing ethical challenges. *The New England journal of medicine*. 2018;378(11):981.

- [13] Armitage H. Researchers create guide for fair and Equitable AI in Health Care. Logo Left ContentLogo Right Content 10,000+ Posts Scope Stanford University School of Medicine blog; 2022. Available from: <https://scopeblog.stanford.edu/2022/10/03/researchers-create-guide-for-fair-and-equitable-ai-in-health-care/>.
- [14] Bedoya AD, Economou-Zavlanos NJ, Goldstein BA, Young A, Jelovsek JE, O'Brien C, et al. A framework for the oversight and local deployment of safe and high-quality prediction models. *Journal of the American Medical Informatics Association*. 2022.
- [15] Reddy S, Allan S, Coghlan S, Cooper P. A governance model for the application of AI in health care. *Journal of the American Medical Informatics Association*. 2020;27(3):491-7.
- [16] Wiens J, Saria S, Sendak M, Ghassemi M, Liu VX, Doshi-Velez F, et al. Do no harm: a roadmap for responsible machine learning for health care. *Nature medicine*. 2019;25(9):1337-40.
- [17] Sendak MP, Balu S, Schulman KA. Barriers to achieving economies of scale in analysis of EHR data. *Applied Clinical Informatics*. 2017;8(03):826-31.
- [18] Sculley D, Holt G, Golovin D, Davydov E, Phillips T, Ebner D, et al. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*. 2015;28.
- [19] Morse KE, Bagley SC, Shah NH. Estimate the hidden deployment cost of predictive models to improve patient care. *Nature medicine*. 2020;26(1):18-9.
- [20] Siwicki B. An epic cognitive computing platform primer; 2021. Available from: <https://www.healthcareitnews.com/news/epic-cognitive-computing-platform-primer>.
- [21] Kashyap S, Morse KE, Patel B, Shah NH. A survey of extant organizational and computational setups for deploying predictive models in health systems. *Journal of the American Medical Informatics Association*. 2021;28(11):2445-50.
- [22] Azure Functions – Serverless Functions in Computing — Microsoft Azure — [azure.microsoft.com/](https://azure.microsoft.com/en-us/products/functions/); [Accessed 19-Jan-2023]. <https://azure.microsoft.com/en-us/products/functions/>.
- [23] ;. Available from: <https://streamlit.io/>.
- [24] Dash S, Shakyawar SK, Sharma M, Kaushik S. Big data in healthcare: management, analysis and future prospects. *Journal of Big Data*. 2019;6(1):1-25.
- [25] Corbin CK, Sung L, Chattopadhyay A, Noshad M, Chang A, Deresinski S, et al. Personalized antibiograms for machine learning driven antibiotic selection. *Communications medicine*. 2022;2(1):1-14.
- [26] Xu S, Hom J, Balasubramanian S, Schroeder LF, Najafi N, Roy S, et al. Prevalence and predictability of low-yield inpatient laboratory diagnostic tests. *JAMA network open*. 2019;2(9):e1910967-7.
- [27] Datta S, Posada J, Olson G, Li W, O'Reilly C, Balraj D, et al. A new paradigm for accelerating clinical data science at Stanford Medicine. *arXiv preprint arXiv:200310534*. 2020.
- [28] BigQuery Enterprise Data warehouse nbsp;— nbsp; google cloud. Google;. Available from: <https://cloud.google.com/bigquery>.
- [29] Krall M. Acceptance and performance by clinicians using an ambulatory electronic medical record in an HMO. In: Proceedings of the Annual Symposium on Computer Application in Medical Care. American Medical Informatics Association; 1995. p. 708.
- [30] Bender D, Sartipi K. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. In: Proceedings of the 26th IEEE international symposium on computer-based medical systems. IEEE; 2013. p. 326-31.
- [31] Barker W, Johnson C. The ecosystem of apps and software integrated with certified health information technology. *Journal of the American Medical Informatics Association*. 2021;28(11):2379-84.
- [32] Klatt TE, Hopp E. Effect of a best-practice alert on the rate of influenza vaccination of pregnant women. *Obstetrics & Gynecology*. 2012;119(2):301-5.
- [33] Ahmed AM, Bose S, Breeden M, Smeds M. Interacting With Best Practice Advisory (BPA) Notifications In Epic Significantly Improves Screening Rates For Abdominal Aortic Aneurysms. *Journal of Vascular Surgery*. 2022;76(4):e75-6.
- [34] Ye C, Wang O, Liu M, Zheng L, Xia M, Hao S, et al. A real-time early warning system for monitoring inpatient mortality risk: prospective study using electronic medical record data. *Journal of medical Internet research*. 2019;21(7):e13719.
- [35] Nemati S, Holder A, Razmi F, Stanley MD, Clifford GD, Buchman TG. An interpretable machine learning model for accurate prediction of sepsis in the ICU. *Critical care medicine*. 2018;46(4):547.

- [36] Saqib M, Sha Y, Wang MD. Early prediction of sepsis in EMR records using traditional ML techniques and deep learning LSTM networks. In: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE; 2018. p. 4038-41.
- [37] Tomašev N, Glorot X, Rae JW, Zielinski M, Askham H, Saraiva A, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*. 2019;572(7767):116-9.
- [38] Keller MS. Take command: cron: Job scheduler. *Linux Journal*. 1999;1999(65es):15-es.
- [39] Lundberg SM, Lee SI. A Unified Approach to Interpreting Model Predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems 30. Curran Associates, Inc.; 2017. p. 4765-74. Available from: <http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>.
- [40] Flynn MS, Jiang SW, Stauffer T, Kwock JT, Nicholas MW. Tracking procedure outcomes using Epic SmartText and Smart-Data Elements following minor dermatologic procedures in the ambulatory setting. *Journal of the American Academy of Dermatology*. 2022.
- [41] Goldberg HS, Paterno MD, Grundmeier RW, Rocha BH, Hoffman JM, Tham E, et al. Use of a remote clinical decision support service for a multicenter trial to implement prediction rules for children with minor blunt head trauma. *International journal of medical informatics*. 2016;87:101-10.
- [42] Bloomfield Jr RA, Polo-Wood F, Mandel JC, Mandl KD. Opening the Duke electronic health record to apps: implementing SMART on FHIR. *International journal of medical informatics*. 2017;99:1-10.
- [43] Feng J, Phillips RV, Malenica I, Bishara A, Hubbard AE, Celi LA, et al. Clinical artificial intelligence quality improvement: towards continual monitoring and updating of AI algorithms in healthcare. *npj Digital Medicine*. 2022;5(1):1-9.
- [44] Schröder T, Schulz M. Monitoring machine learning models: a categorization of challenges and methods. *Data Science and Management*. 2022;5(3):105-16.
- [45] Klaise J, Van Looveren A, Cox C, Vacanti G, Coca A. Monitoring and explainability of models in production. *arXiv preprint arXiv:200706299*. 2020.
- [46] Jung K, Shah NH. Implications of non-stationarity on predictive modeling using EHRs. *Journal of biomedical informatics*. 2015;58:168-74.
- [47] Vickers AJ, Van Calster B, Steyerberg EW. Net benefit approaches to the evaluation of prediction models, molecular markers, and diagnostic tests. *bmj*. 2016;352.
- [48] Tonekaboni S, Morgenshtern G, Assadi A, Pokhrel A, Huang X, Jayarajan A, et al. How to validate Machine Learning Models Prior to Deployment: Silent trial protocol for evaluation of real-time models at ICU. In: Conference on Health, Inference, and Learning. PMLR; 2022. p. 169-82.
- [49] Krautbacher N, Theis FJ, Fuchs C. Correcting classifiers for sample selection bias in two-phase case-control studies. *Computational and mathematical methods in medicine*. 2017;2017.
- [50] Reps JM, Ryan PB, Rijnbeek PR, Schuemie MJ. Design matters in patient-level prediction: evaluation of a cohort vs. case-control design when developing predictive models in observational healthcare datasets. *Journal of Big Data*. 2021;8(1):1-18.
- [51] David G, Smith-McLallen A, Ukert B. The effect of predictive analytics-driven interventions on healthcare utilization. *Journal of health economics*. 2019;64:68-79.
- [52] Nemati S, Shashikumar SP, Holder AL, Wardi G, Owens RL. Randomized Clinical Trials or Convenient Controls: TREWS or FALSE? *medRxiv*. 2022.
- [53] Escobar GJ, Liu VX, Schuler A, Lawson B, Greene JD, Kipnis P. Automated identification of adults at risk for in-hospital clinical deterioration. *New England Journal of Medicine*. 2020;383(20):1951-60.
- [54] Aikens RC, Balasubramanian S, Chen JH. A machine learning approach to predicting the stability of inpatient lab test results. *AMIA Summits on Translational Science Proceedings*. 2019;2019:515.
- [55] Rabbani N, Ma SP, Li RC, Winget M, Weber S, Boosi S, et al. Targeting Repetitive Laboratory Testing with Electronic Health Records-Embedded Predictive Decision Support: A Pre-Implementation Study. *Clinical Biochemistry*. 2023.
- [56] Kim GY, Noshad M, Stehr H, Rojansky R, Gratzinger D, Oak J, et al. Machine Learning Predictability of Clinical Next Generation Sequencing for Hematologic Malignancies to Guide High-Value Precision Medicine. In: AMIA Annual Symposium Proceedings. vol. 2021. American Medical Informatics Association; 2021. p. 641.
- [57] Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, et al. Scalable and accurate deep learning with electronic health records. *NPJ digital medicine*. 2018;1(1):1-10.

- [58] Lewin-Epstein O, Baruch S, Hadany L, Stein GY, Obolski U. Predicting antibiotic resistance in hospitalized patients by applying machine learning to electronic medical records. *Clinical Infectious Diseases*. 2021;72(11):e848-55.
- [59] Corbin CK, Medford RJ, Osei K, Chen JH. Personalized antibiograms: machine learning for precision selection of empiric antibiotics. *AMIA Summits on Translational Science Proceedings*. 2020;2020:108.
- [60] Eanoff S, Obermeyer Z, Butte AJ. The case for algorithmic stewardship for artificial intelligence and machine learning technologies. *Jama*. 2020;324(14):1397-8.
- [61] Yelin I, Snitser O, Novich G, Katz R, Tal O, Parizade M, et al. Personal clinical history predicts antibiotic resistance of urinary tract infections. *Nature medicine*. 2019;25(7):1143-52.
- [62] Kanjilal S, Oberst M, Boominathan S, Zhou H, Hooper DC, Sontag D. A decision algorithm to promote outpatient antimicrobial stewardship for uncomplicated urinary tract infection. *Science Translational Medicine*. 2020;12(568):eaay5067.
- [63] Nguyen M, Corbin CK, Eulalio T, Ostberg NP, Machiraju G, Marafino BJ, et al. Developing machine learning models to personalize care levels among emergency room patients for hospital admission. *Journal of the American Medical Informatics Association*. 2021;28(11):2423-32.
- [64] Chen JH, Podchiyska T, Altman RB. OrderRex: clinical order decision support and outcome predictions by data-mining electronic medical records. *Journal of the American Medical Informatics Association*. 2016;23(2):339-48.
- [65] Chen JH, Altman RB. Automated physician order recommendations and outcome predictions by data-mining electronic medical records. *AMIA Summits on Translational Science proceedings*. 2014;2014:206.
- [66] Ip W, Prahalad P, Palma J, Chen JH, et al. A Data-Driven Algorithm to Recommend Initial Clinical Workup for Outpatient Specialty Referral: Algorithm Development and Validation Using Electronic Health Record Data and Expert Surveys. *JMIR Medical Informatics*. 2022;10(3):e30104.
- [67] Kim JS, Vivas A, Arvind V, Lombardi J, Reidler J, Zuckerman SL, et al. Can natural language processing and artificial intelligence automate the generation of billing codes from operative note dictations? *Global Spine Journal*. 2022;21925682211062831.
- [68] Bajor JM, Lasko TA. Predicting medications from diagnostic codes with recurrent neural networks. In: International conference on learning representations; 2017. .
- [69] Schäfer H, Friedrich CM. UMLS mapping and Word embeddings for ICD code assignment using the MIMIC-III intensive care database. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE; 2019. p. 6089-92.
- [70] Adams R, Henry KE, Sridharan A, Soleimani H, Zhan A, Rawat N, et al. Prospective, multi-site study of patient outcomes after implementation of the TREWS machine learning-based early warning system for sepsis. *Nature medicine*. 2022;28(7):1455-60.
- [71] Rabanser S, Günnemann S, Lipton Z. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*. 2019;32.
- [72] Davis SE, Greevy Jr RA, Lasko TA, Walsh CG, Matheny ME. Detection of calibration drift in clinical prediction models to inform model updating. *Journal of biomedical informatics*. 2020;112:103611.
- [73] Davis SE, Greevy Jr RA, Fonnesbeck C, Lasko TA, Walsh CG, Matheny ME. A nonparametric updating method to correct clinical prediction model drift. *Journal of the American Medical Informatics Association*. 2019;26(12):1448-57.
- [74] Lenert MC, Matheny ME, Walsh CG. Prognostic models will be victims of their own success, unless.... *Journal of the American Medical Informatics Association*. 2019;26(12):1645-50.
- [75] Perdomo J, Zrnic T, Mendler-Dünner C, Hardt M. Performative prediction. In: International Conference on Machine Learning. PMLR; 2020. p. 7599-609.
- [76] Adam GA, Chang CHK, Haibe-Kains B, Goldenberg A. Hidden risks of machine learning applied to healthcare: unintended feedback loops between models and future data causing model degradation. In: Machine Learning for Healthcare Conference. PMLR; 2020. p. 710-31.
- [77] Adam GA, Chang CHK, Haibe-Kains B, Goldenberg A. Error amplification when updating deployed machine learning models. In: Proceedings of the Machine Learning for Healthcare Conference, Durham, NC, USA; 2022. p. 5-6.
- [78] Corbin CK, Baiocchi M, Chen JH. Avoiding Biased Clinical Machine Learning Model Performance Estimates in the Presence of Label Selection. *arXiv preprint arXiv:220909188*. 2022.
- [79] Gordon WJ, Mandl KD. The 21st century cures act: a competitive apps market and the risk of innovation blocking. *Journal of Medical Internet Research*. 2020;22(12):e24824.

- [80] Rakha EA, Toss M, Shiino S, Gamble P, Jaroensri R, Mermel CH, et al. Current and future applications of artificial intelligence in pathology: a clinical perspective. *Journal of clinical pathology*. 2021;74(7):409-14.
- [81] Rezazade Mehrizi MH, van Ooijen P, Homan M. Applications of artificial intelligence (AI) in diagnostic radiology: a technography study. *European radiology*. 2021;31:1805-11.
- [82] Eapen BR. Artificial intelligence in dermatology: A practical introduction to a paradigm shift. *Indian dermatology online journal*. 2020;11(6):881.
- [83] Blueprint for trustworthy AI implementation guidance and assurance for healthcare;. [Accessed 7-March-2023]. <https://tinyurl.com/CHAI-paper>.

Supplemental Note 1

We created feature matrices for each cohort using count based representations⁵⁷. For each cohort, a timeline of medical events was constructed from structured electronic medical record data available before inference time. A mix of categorical and numerical data elements were considered. Categorical features included diagnosis (ICD 10) codes on a patient's problem list, medication orders and demographic variables including race and sex. Numerical features included prior laboratory results and the patient's age at inference time. Numerical features were discretized into tokens based on the percentile values they assumed in the training set distribution. All numerical features were binned into five buckets. All diagnosis codes prior to prediction time were included in the patient's constructed timeline. Medication orders placed within 28 days of prediction time were included, as were laboratory results made available within 14 days of prediction time. Sequences of tokens were then transformed into feature vectors in bag of words (counts) fashion. The total number of features was 14145, 13631, and 14917 for the CBC, metabolic panel, and magnesium cohorts respectively. Each cohort was split into training, validation and test sets based on the year the diagnostic order took place. Training sets included the years 2015 to 2019. Validation sets included orders taking place in 2020, and test sets included orders taking place in 2021. Random forest models were trained using the training sets, hyperparameters chosen using the validation sets, and performance evaluated on the test sets. We used the scikit-learn implementation of random forest models for these examples, though DEPLOYR allows any arbitrary model class or package.

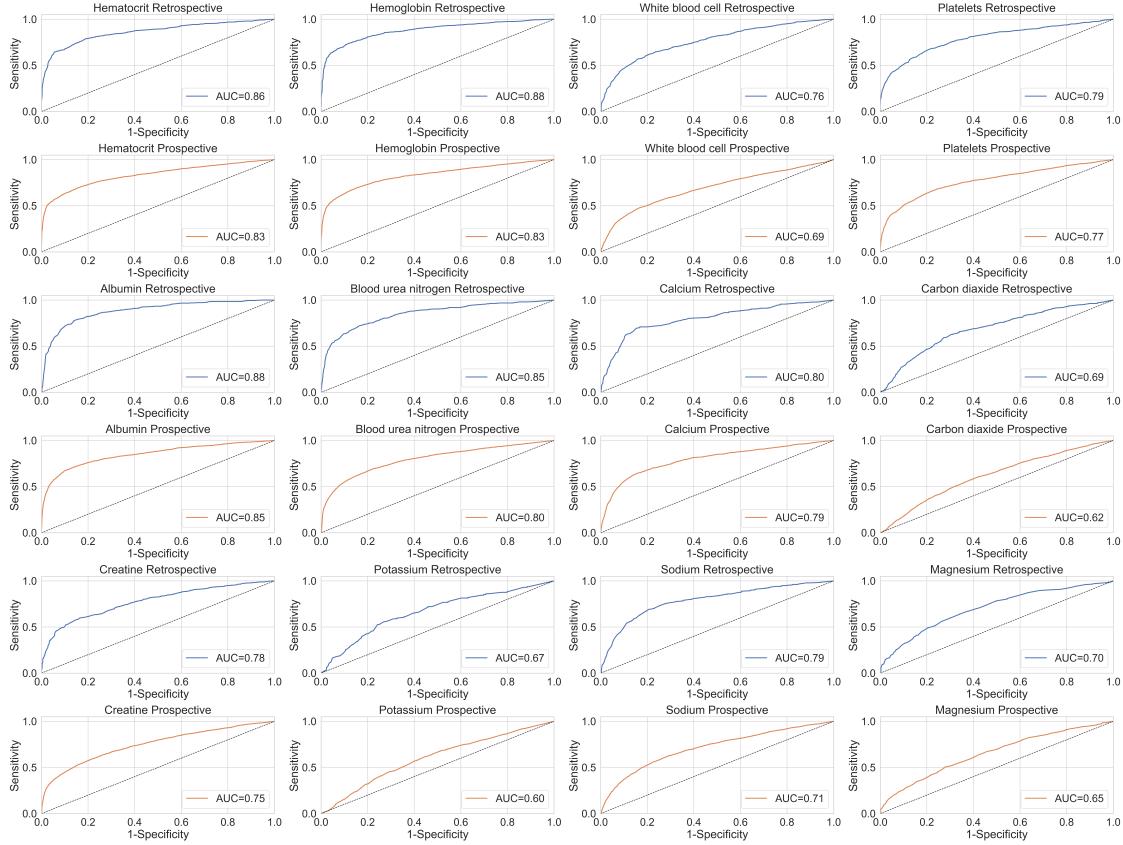


Figure 6: Receiver operating characteristic curves for twelve deployed models on retrospective and prospective test sets.

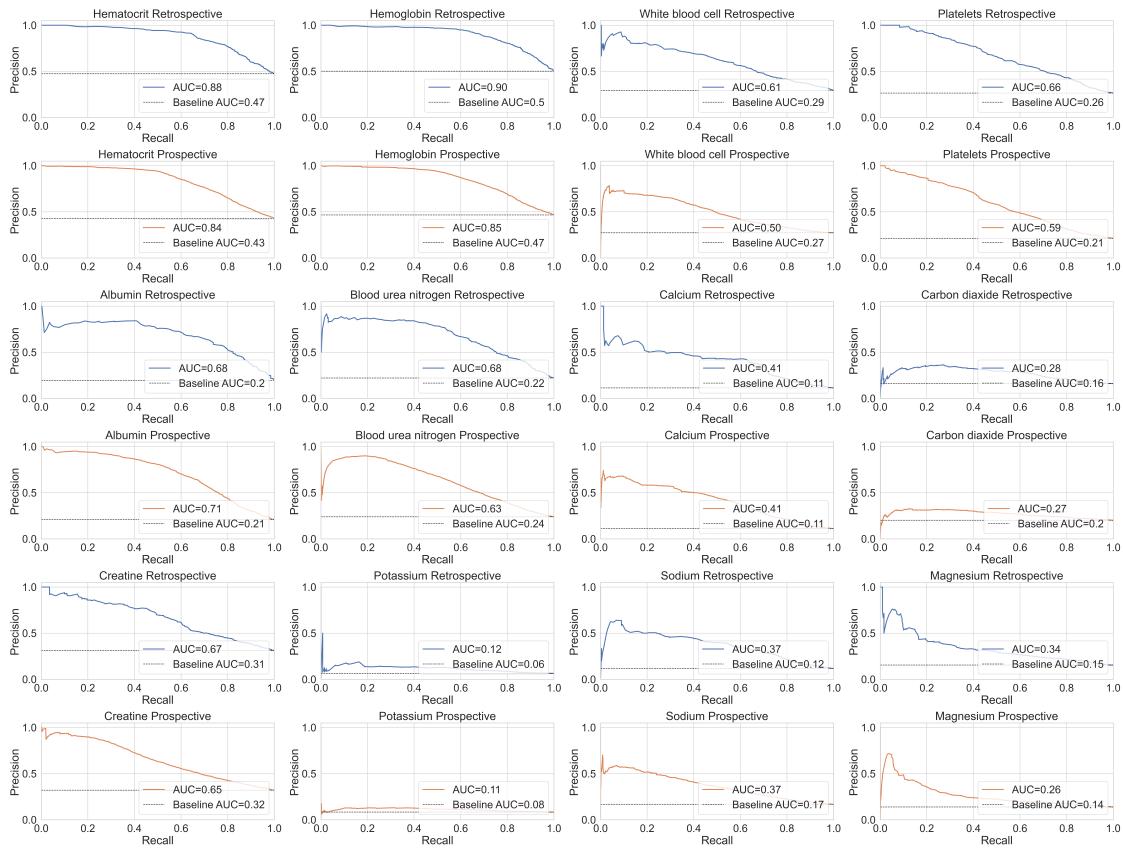


Figure 7: Precision recall curves for twelve deployed models on retrospective and prospective test sets.

Table 3: Hematocrit model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Hematocrit	Full cohort	0.86 [0.85, 0.88]	0.83 [0.83, 0.84]
	sex_Female	0.85 [0.83, 0.88]	0.83 [0.82, 0.84]
	sex_Male	0.87 [0.84, 0.89]	0.83 [0.83, 0.84]
	race_Asian	0.85 [0.81, 0.88]	0.85 [0.83, 0.86]
	race_Black	0.84 [0.75, 0.91]	0.79 [0.76, 0.81]
	race_Native American	0.87 [0.55, 1.00]	0.81 [0.70, 0.90]
	race_Other	0.85 [0.81, 0.89]	0.82 [0.81, 0.83]
	race_Pacific Islander	0.91 [0.70, 1.00]	0.85 [0.81, 0.89]
	race_Unknown	0.80 [0.66, 0.90]	0.75 [0.68, 0.81]
	race_White	0.88 [0.85, 0.90]	0.84 [0.83, 0.85]
	age_over_40	0.87 [0.85, 0.89]	0.84 [0.83, 0.85]
	sex_Unknown	NaN	1.00 [1.00, 1.00]

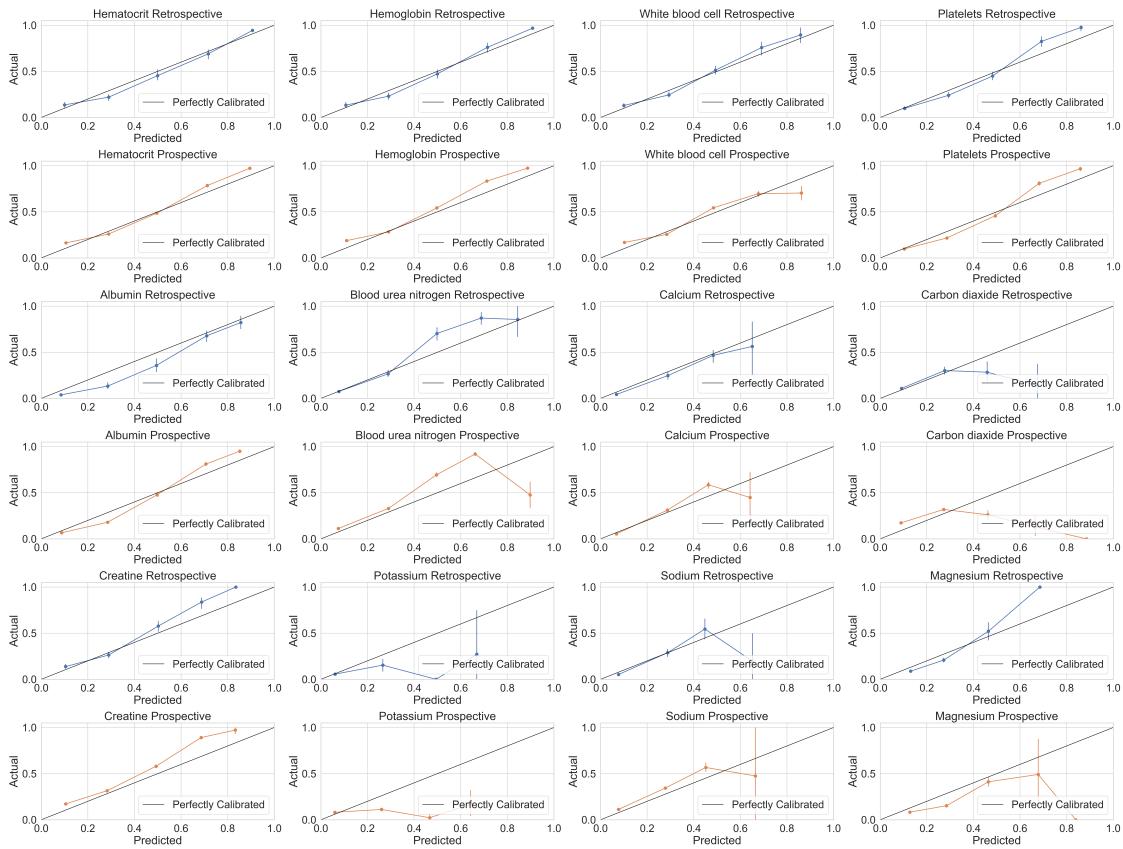


Figure 8: Calibration plots for twelve deployed models on retrospective and prospective test sets.

Table 4: Hemoglobin model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Hemoglobin	Full cohort	0.88 [0.86, 0.89]	0.83 [0.83, 0.84]
	sex_Female	0.88 [0.86, 0.90]	0.82 [0.81, 0.83]
	sex_Male	0.88 [0.86, 0.90]	0.84 [0.83, 0.85]
	race_Asian	0.87 [0.84, 0.90]	0.85 [0.84, 0.86]
	race_Black	0.85 [0.77, 0.93]	0.77 [0.74, 0.80]
	race_Native American	1.00 [1.00, 1.00]	0.81 [0.71, 0.90]
	race_Other	0.87 [0.84, 0.91]	0.82 [0.80, 0.83]
	race_Pacific Islander	0.85 [0.59, 1.00]	0.84 [0.80, 0.88]
	race_Unknown	0.80 [0.67, 0.90]	0.70 [0.63, 0.76]
	race_White	0.89 [0.87, 0.91]	0.84 [0.84, 0.85]
	age_over_40	0.89 [0.87, 0.91]	0.84 [0.83, 0.85]
	sex_Unknown	NaN	1.00 [1.00, 1.00]

Table 5: White blood cell model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
White blood cell	<i>Full cohort</i>	0.76 [0.74, 0.79]	0.69 [0.68, 0.70]
	sex_Female	0.77 [0.73, 0.80]	0.68 [0.67, 0.69]
	sex_Male	0.76 [0.73, 0.80]	0.70 [0.69, 0.71]
	race_Asian	0.76 [0.70, 0.82]	0.70 [0.68, 0.72]
	race_Black	0.75 [0.63, 0.86]	0.68 [0.64, 0.72]
	race_Native American	0.50 [0.13, 0.88]	0.68 [0.49, 0.82]
	race_Other	0.73 [0.68, 0.78]	0.67 [0.65, 0.69]
	race_Pacific Islander	1.00 [1.00, 1.00]	0.76 [0.69, 0.83]
	race_Unknown	0.87 [0.74, 0.96]	0.69 [0.60, 0.76]
	race_White	0.77 [0.73, 0.80]	0.70 [0.68, 0.71]
	age_over_40	0.76 [0.73, 0.79]	0.70 [0.69, 0.71]
	sex_Unknown	NaN	0.80 [0.40, 1.00]

Table 6: Platelets model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Platelets	<i>Full cohort</i>	0.79 [0.77, 0.82]	0.77 [0.76, 0.78]
	sex_Female	0.80 [0.77, 0.84]	0.76 [0.75, 0.77]
	sex_Male	0.78 [0.74, 0.81]	0.78 [0.77, 0.79]
	race_Asian	0.78 [0.72, 0.84]	0.79 [0.77, 0.81]
	race_Black	0.85 [0.75, 0.93]	0.76 [0.72, 0.79]
	race_Native American	0.89 [0.67, 1.00]	0.76 [0.60, 0.90]
	race_Other	0.76 [0.70, 0.81]	0.75 [0.73, 0.77]
	race_Pacific Islander	0.69 [0.07, 1.00]	0.77 [0.69, 0.85]
	race_Unknown	0.87 [0.72, 0.98]	0.64 [0.51, 0.74]
	race_White	0.80 [0.77, 0.83]	0.78 [0.76, 0.79]
	age_over_40	0.78 [0.75, 0.81]	0.77 [0.76, 0.78]
	sex_Unknown	NaN	NaN

Table 7: Albumin model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Albumin	<i>Full cohort</i>	0.88 [0.86, 0.91]	0.85 [0.84, 0.86]
	sex_Female	0.89 [0.86, 0.91]	0.86 [0.85, 0.87]
	sex_Male	0.88 [0.86, 0.91]	0.84 [0.83, 0.85]
	race_Asian	0.81 [0.74, 0.87]	0.84 [0.82, 0.86]
	race_Black	0.91 [0.80, 0.97]	0.83 [0.78, 0.87]
	race_Native American	0.93 [0.61, 1.00]	0.80 [0.69, 0.90]
	race_Other	0.88 [0.83, 0.91]	0.84 [0.82, 0.86]
	race_Pacific Islander	0.73 [0.46, 0.96]	0.89 [0.83, 0.93]
	race_Unknown	0.84 [0.63, 1.00]	0.83 [0.75, 0.91]
	race_White	0.91 [0.89, 0.93]	0.86 [0.85, 0.87]
	age_over_40	0.89 [0.87, 0.91]	0.87 [0.86, 0.88]
	sex_Unknown	NaN	NaN

Table 8: Blood urea nitrogen model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Blood urea nitrogen	<i>Full cohort</i>	0.85 [0.83, 0.87]	0.80 [0.79, 0.81]
	sex_Female	0.84 [0.81, 0.88]	0.79 [0.77, 0.80]
	sex_Male	0.85 [0.82, 0.87]	0.80 [0.78, 0.81]
	race_Asian	0.86 [0.80, 0.92]	0.81 [0.79, 0.83]
	race_Black	0.85 [0.72, 0.95]	0.82 [0.78, 0.85]
	race_Native American	1.00 [1.00, 1.00]	0.91 [0.82, 0.97]
	race_Other	0.85 [0.80, 0.88]	0.78 [0.76, 0.80]
	race_Pacific Islander	0.91 [0.75, 1.00]	0.83 [0.77, 0.88]
	race_Unknown	0.85 [0.61, 0.99]	0.74 [0.67, 0.82]
	race_White	0.84 [0.81, 0.87]	0.79 [0.78, 0.80]
	age_over_40	0.85 [0.82, 0.87]	0.79 [0.78, 0.80]
	sex_Unknown	NaN	NaN

Table 9: Calcium model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Calcium	<i>Full cohort</i>	0.80 [0.76, 0.83]	0.79 [0.78, 0.81]
	sex_Female	0.76 [0.70, 0.81]	0.78 [0.76, 0.80]
	sex_Male	0.83 [0.79, 0.87]	0.81 [0.79, 0.82]
	race_Asian	0.76 [0.66, 0.86]	0.78 [0.76, 0.81]
	race_Black	0.80 [0.69, 0.91]	0.75 [0.68, 0.81]
	race_Native American	0.78 [0.33, 1.00]	0.81 [0.68, 0.93]
	race_Other	0.84 [0.78, 0.89]	0.82 [0.80, 0.84]
	race_Pacific Islander	0.87 [0.73, 1.00]	0.81 [0.71, 0.89]
	race_Unknown	0.64 [0.36, 0.89]	0.78 [0.61, 0.93]
	race_White	0.78 [0.73, 0.84]	0.79 [0.77, 0.81]
	age_over_40	0.79 [0.75, 0.83]	0.79 [0.78, 0.81]
	sex_Unknown	NaN	NaN

Table 10: Carbon dioxide model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Carbon dioxide	<i>Full cohort</i>	0.69 [0.66, 0.72]	0.62 [0.61, 0.63]
	sex_Female	0.67 [0.62, 0.71]	0.61 [0.59, 0.62]
	sex_Male	0.71 [0.66, 0.76]	0.63 [0.61, 0.64]
	race_Asian	0.73 [0.64, 0.80]	0.61 [0.59, 0.64]
	race_Black	0.76 [0.63, 0.88]	0.62 [0.58, 0.66]
	race_Native American	NaN	0.74 [0.61, 0.84]
	race_Other	0.72 [0.65, 0.77]	0.63 [0.60, 0.65]
	race_Pacific Islander	0.26 [0.04, 0.54]	0.69 [0.62, 0.76]
	race_Unknown	0.96 [0.91, 1.00]	0.64 [0.55, 0.73]
	race_White	0.65 [0.60, 0.70]	0.61 [0.59, 0.62]
	age_over_40	0.69 [0.65, 0.72]	0.62 [0.61, 0.64]
	sex_Unknown	NaN	0.75 [0.25, 1.00]

Table 11: Creatinine model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Creatinine	<i>Full cohort</i>	0.78 [0.75, 0.80]	0.75 [0.74, 0.76]
	sex_Female	0.76 [0.73, 0.80]	0.74 [0.73, 0.75]
	sex_Male	0.77 [0.74, 0.81]	0.75 [0.74, 0.76]
	race_Asian	0.79 [0.74, 0.85]	0.75 [0.73, 0.77]
	race_Black	0.85 [0.77, 0.93]	0.76 [0.73, 0.80]
	race_Native American	1.00 [1.00, 1.00]	0.93 [0.87, 0.98]
	race_Other	0.80 [0.75, 0.84]	0.72 [0.71, 0.74]
	race_Pacific Islander	0.67 [0.43, 0.88]	0.81 [0.75, 0.86]
	race_Unknown	0.62 [0.43, 0.80]	0.71 [0.64, 0.78]
	race_White	0.75 [0.72, 0.79]	0.75 [0.74, 0.76]
	age_over_40	0.78 [0.75, 0.80]	0.76 [0.75, 0.77]
	sex_Unknown	NaN	0.88 [0.50, 1.00]

Table 12: Potassium model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Potassium	<i>Full cohort</i>	0.67 [0.61, 0.72]	0.60 [0.59, 0.62]
	sex_Female	0.65 [0.57, 0.72]	0.60 [0.58, 0.62]
	sex_Male	0.68 [0.62, 0.74]	0.60 [0.58, 0.63]
	race_Asian	0.75 [0.66, 0.83]	0.61 [0.57, 0.65]
	race_Black	0.79 [0.66, 0.90]	0.56 [0.50, 0.62]
	race_Native American	NaN	0.59 [0.46, 0.72]
	race_Other	0.64 [0.55, 0.74]	0.58 [0.55, 0.61]
	race_Pacific Islander	0.45 [0.06, 0.87]	0.56 [0.47, 0.64]
	race_Unknown	0.32 [0.10, 0.82]	0.64 [0.46, 0.80]
	race_White	0.65 [0.57, 0.72]	0.61 [0.58, 0.63]
	age_over_40	0.65 [0.59, 0.71]	0.60 [0.59, 0.62]
	sex_Unknown	NaN	0.00 [0.00, 0.00]

Table 13: Sodium model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Sodium	<i>Full cohort</i>	0.79 [0.75, 0.82]	0.71 [0.70, 0.72]
	sex_Female	0.78 [0.73, 0.83]	0.71 [0.69, 0.72]
	sex_Male	0.79 [0.74, 0.83]	0.72 [0.70, 0.73]
	race_Asian	0.81 [0.73, 0.87]	0.73 [0.70, 0.75]
	race_Black	0.78 [0.65, 0.89]	0.71 [0.65, 0.75]
	race_Native American	0.73 [0.14, 1.00]	0.74 [0.59, 0.89]
	race_Other	0.78 [0.72, 0.84]	0.68 [0.65, 0.70]
	race_Pacific Islander	0.91 [0.81, 1.00]	0.70 [0.62, 0.78]
	race_Unknown	0.72 [0.45, 0.99]	0.75 [0.65, 0.84]
	race_White	0.78 [0.72, 0.83]	0.72 [0.71, 0.74]
	age_over_40	0.77 [0.73, 0.81]	0.71 [0.70, 0.72]
	sex_Unknown	NaN	0.20 [0.00, 0.60]

Table 14: Magnesium model performance by protected demographic groups

Prediction task	Group	Retrospective AUROC	Prospective AUROC
Magnesium	<i>Full cohort</i>	0.70 [0.67, 0.73]	0.65 [0.63, 0.67]
	sex_Female	0.71 [0.66, 0.75]	0.67 [0.63, 0.70]
	sex_Male	0.70 [0.66, 0.74]	0.64 [0.61, 0.67]
	sex_Unknown	NaN	NaN
	race_Asian	0.76 [0.66, 0.84]	0.66 [0.61, 0.72]
	race_Black	0.69 [0.53, 0.82]	0.70 [0.62, 0.77]
	race_Native American	NaN	0.53 [0.29, 0.75]
	race_Other	0.69 [0.62, 0.74]	0.65 [0.61, 0.70]
	race_Pacific Islander	0.86 [0.67, 0.99]	0.60 [0.47, 0.72]
	race_Unknown	0.64 [0.28, 0.94]	0.42 [0.18, 0.67]
	race_White	0.70 [0.65, 0.74]	0.65 [0.62, 0.68]
	age_over_40	0.70 [0.66, 0.74]	0.65 [0.63, 0.68]