Developing Continuous Risk Models for Adverse Event Prediction in Electronic Health Records using Deep Learning

Nenad Tomašev[1], Natalie Harris[2], Sebastien Baur[2], Anne Mottram[2], Xavier Glorot[1], Jack W. Rae[1, 3], Michal Zielinski[1], Harry Askham[1], Andre Saraiva[1], Valerio Magliulo[2], Clemens Meyer[1], Suman Ravuri[1], Ivan Protsyuk[2], Alistair Connell[2], Cían O. Hughes[2], Alan Karthikesalingam[2], Julien Cornebise[1, 4], Hugh Montgomery[5], Geraint Rees[6], Chris Laing[7], Clifton R. Baker[8], Thomas F. Osborne[10, 11], Ruth Reeves[8], Demis Hassabis[1], Dominic King[2], Mustafa Suleyman[1], Trevor Back[1], Christopher Nielson[8, 13,14], Martin G. Seneviratne[2,14], Joseph R. Ledsam[1,14,*], and Shakir Mohamed[1,14]

[1]DeepMind, London, UK

[2]Google Health, London, UK

[3]CoMPLEX, Computer Science, University College London, London, UK

[4]Present address: University College London, London, UK

[5]Institute for Human Health and Performance, University College London, London, UK

[6]Institute of Cognitive Neuroscience, University College London, London, UK

[7]University College London Hospitals, London, UK

[8]Department of Veterans Affairs, USA

[9]VA Salt Lake City Healthcare System, USA

[10]VA Palo Alto Healthcare System, USA

[11]Stanford University School of Medicine, USA

[12]Division of Epidemiology, University of Utah, USA

[13]University of Nevada School of Medicine, USA

[14]These authors contributed equally to this work

[*]email:jledsam@google.com

Abstract

Early prediction of patient outcomes is key to unlocking the potential for targeted preventive care. This protocol describes a practical workflow for developing deep learning risk models for early prediction of various clinical and operational outcomes using structured electronic health record (EHR) data, discussing the prediction of acute kidney injury (AKI) as an exemplar. The protocol consists of 34 steps grouped into the following stages: formal problem definition, data pre-processing, architecture selection, calibration and uncertainty estimation, generalisability evaluation. Additionally, we demonstrate the application of this protocol to three other endpoints - mortality, length of stay and 30-day hospital readmission - for both continuous predictions (e.g. triggered every 6h) and static predictions (e.g. triggered at 24h post admission). The performance on these additional endpoints exceeded most comparable literature benchmarks. This protocol is accompanied by an open-source codebase that illustrates key considerations for EHR modeling and may be customised to alternate data formats and prediction tasks.

Keywords: machine learning, deep learning, artificial intelligence, electronic health records, risk prediction, acute kidney injury, mortality, length of stay, readmission

1 Introduction

Early prediction of patient outcomes is a major focus of clinical quality guidelines for deterioration [1], sepsis [2], acute kidney injury (AKI) [3], hospital readmissions [4], etc. Quality improvement initiatives aiming to deliver anticipatory care often consist of two components: (i) an afferent limb, to identify high-risk patients (e.g. early warning scores); and (ii) an efferent limb, to respond to that alert (e.g. clinical outreach team) [5]. Traditionally, the afferent limb involves a rule-based patient risk score - for example, the National Early Warning Score (NEWS2) and Modified Early Warning Score (MEWS) for acute deterioration [6, 7] or the LACE and HOSPITAL scores for readmission [8, 9]. Rule-based scores are limited in that they are typically not personalised to the patient (with the same thresholds and coefficients used population-wide),

rarely use temporal or trend information, have a relatively limited input feature space, and have been associated with high false positive rates and consequent alert fatigue [10, 11].

The emerging opportunity to run machine learning (ML) models on continuously streamed electronic health record (EHR) data may help to tackle some of the above limitations. ML models have been developed to predict a wide array of clinical and operational outcomes, in order to risk-stratify patients and inform treatment decisions. Use cases have ranged from mortality and length of stay prediction to more targeted algorithms for AKI, sepsis, shock or delirium, with an increasing focus on deep learning approaches [12–34].

To date, few of these models have been validated prospectively and adopted in routine practice [35, 36]; however implementation studies are beginning to emerge [37]. As ML systems start to infiltrate clinical practice, it is critical that the upstream data-science pipelines for developing and evaluating deep learning models with EHR data are robust and well specified. Several guidelines have recently been published about the development of ML models in healthcare, covering key principles such as problem selection, fairness, bias, model surveillance and outcomes evaluation across various data modalities [38–43]. In addition, there are initiatives to create standardised reporting guidelines for ML studies using clinical data, including the TRIPOD-ML framework [44, 45]. However, few systematic protocols exist for the development of deep learning models using EHR data, detailing the practical steps and challenges of training risk models using retrospective data.

In this protocol, we outline a clear workflow for developing supervised deep learning models on structured EHR data (excluding free-text clinical notes). The AKI prediction model introduced in Tomasev $etal.$ [12] is used as the primary exemplar; however the protocol can be applied to a wide range of clinical and/or operational use cases and is demonstrated on three additional endpoints: mortality, length of stay and hospital readmission.

1.1 Development of the protocol

The protocol references the clinical use case of AKI prediction, as described in our previous work [12]. The AKI prediction model was developed using a large EHR dataset from the US Department of Veterans Affairs (VA) [46], consisting of de-identified longitudinal data on 703,782 adult patients across 172 inpatient and 1,062 outpatient sites. Inclusion criteria were patients aged between 18 and 90 years admitted for secondary care to medical or surgical services from the beginning of October 2011 to the end of September 2015. Due to the patient population of the VA, the dataset consisted of 93.6% male subjects. The dataset included laboratory tests, vital signs, medications, admissions, transfers, outpatient visits, diagnoses as International Classification of Diseases (ICD9) codes and procedures as Current Procedural Terminology (CPT) codes. All patient data were de-identified. Additional precautions beyond standard de-identification were taken to safeguard patient privacy: free text notes and rare diagnoses were excluded; many feature names were obfuscated (i.e. the feature value was preserved but the name was obfuscated); and all patient records were time-jittered, respecting relative temporal relationships for individual patients. A more comprehensive dataset description is provided in the Methods and Extended Data Table 6 of [12]. This protocol implicitly assumes that the raw EHR data has been extracted in tabular format from a research data warehouse, with feature names and corresponding continuous or discrete values. Standard EHR data models, e.g. Fast Healthcare Interoperability Resources (FHIR) or the Observational Medical Outcomes Partnership (OMOP) data model could be converted to tabular format or embedded as a vector representation as per [14].

The protocol involves five broad stages: (a) formal problem definition, (b) data preprocessing, (c) architecture selection, (d) calibration and uncertainty estimation, and (e) model generalisability evaluation. The protocol is intended to be applicable to a range of prediction targets. Here we present novel results on several non-AKI endpoints: mortality, length of stay and readmission. These endpoints were chosen because they could be reliably identified in the current dataset and numerous ML benchmarks were available in the literature. Additionally, mortality and readmission have been identified as key operational use cases where analytics could

reduce patient harm and healthcare cost [47]. We also present results for alternate temporal configurations including different triggering schemes (i.e. static predictions versus continuous or regularly-triggered predictions) and different lookahead windows. The protocol could feasibly be generalised to other targets over acute and chronic timescales, such as sepsis, ICU transfer or chronic disease progression.

The protocol consists of 34 steps. While no individual step is methodologically novel, the steps related to auxiliary multi-tasking, interval masking, architecture ablation, uncertainty estimation and clinically-motivated operating points are not in widespread use in the EHR literature. The protocol is accompanied by an open-sourced codebase which illustrates many of these key components. Although customisation is required to use this protocol on new EHR datasets and tasks, we believe this is a useful high-level framework that addresses some of the nuances of supervised learning with EHRs.

1.2 Comparison with other methods

Protocols for developing risk prediction models have previously been proposed [38, 40, 44, 48]. These works tend to describe high-level principles in algorithm design and implementation; whereas our protocol details an explicit methodology for developing deep learning models tailored to longitudinal EHR data, with practical guidance around data pre-processing, architecture selection, hyperparameter sweeps, post-processing and evaluation.

Steyerberg *et al.* [48] provide a useful framework including seven steps for model development and four for model validation; however the framework is focused on regression models at a single time-point with a much smaller input feature space, and does not address the nuances of deep learning with large EHR datasets. Chen *et al.* provide a rigorous review of the steps involved in training and evaluating deep learning models [38]; however our protocol provides more granular detail about the step-by-step approach for dealing with structured EHR data and showcases its application to four distinct prediction targets.

There is a wealth of recent work describing the development of deep learning models for

various EHR endpoints, which follow the steps outlined in our protocol to varying degrees [13–29, 31–34, 49]. While direct comparison of model performance against these prior works is challenging since different datasets are used and experimental setups for the same task can differ (in terms of endpoint definition, lookahead window, triggering frequency, etc), we endeavour to compare our results for mortality, length of stay and readmission with previous ML benchmarks (Section 4).

1.3 Expertise needed to implement the protocol

Successful application of the protocol requires interdisciplinary collaboration. Some steps call for significant technical expertise in deep learning and should be executed by researchers with statistical and ML skills; while others require clinical or operational knowledge and should be executed by clinicians or informaticians. Prospective evaluation calls for experts in trial design and outcomes evaluation. As a guide, steps 1-6, 8, 15-16, 24, 29- 30 and 34 are likely to be clinician-led, while steps 7, 9-14, 17-23, 25-28, 31-33 are likely to be engineer-led.

1.4 Limitations

The dataset from [12] was curated for the purpose of AKI prediction. Additional measures beyond de-identification were taken to preserve privacy, including obfuscation of many feature names and jittering of continuous variables. In order to demonstrate the generalisability of the protocol to other endpoints, we showcase models for mortality, length of stay and readmission; however the selection of auxiliary tasks and the input features for the baseline models were constrained by the panel of named features available for the AKI study (the remaining feature names were obfuscated but values preserved). Performance is still comparable with literature benchmarks, however these models should be considered as prototypes for demonstrating methods rather than clinical-grade ML systems.

The dataset used here was from a diverse range of health facilities and geographies within the VA network, spanning over 5 years; however was limited by a lack of gender diversity due to the

predominantly male patient population. Furthermore, the protocol does not explicitly deal with fairness evaluation. Principles of ML fairness should infuse the entire protocol from problem conception and dataset choice to eval strategy and deployment, however we refer to previous works for a more formal discussion [50, 51].

The protocol is intended to illustrate modeling considerations agnostic to EHR data format. Although we provide exemplar data at different stages of pre-processing, this is not intended as a canonical data representation. We refer the reader to alternate works describing how standard EHR data formats such as FHIR may be embedding into a vector representation [14], to which this protocol could be applied.

This protocol does not support using unstructured text in clinical notes, which constitutes a significant portion of the information content of the EHR [52]. The main reason for the exclusion is that the research dataset used for model development in [12] did not include clinical notes. The protocol could be extended with a natural language processing (NLP) component, which could either be pre-trained and fine-tuned, or trained end-to-end along with the risk model [53].

Finally, the protocol does not currently provide any causal guarantees, allowing only for associative modelling between input features and outcome targets. We emphasise that causal inference using observational EHR data is an important area of active research, which stands to assist in knowledge discovery, ML robustness and fairness [26, 54, 55].

2 Materials

Execution of the protocol requires access to data, computational resources, and relevant software packages for data analysis and machine learning. Here we list the key software packages used in [12]; however there are many alternative infrastructure options available.

2.1 Software

- Data processing framework **Apache Beam** (`https://beam.apache.org/`)

- Plotting library **Matplotlib** [56] (https://matplotlib.org/)
- Scientific computing library **Numpy** [57] (https://www.numpy.org/)
- Scientific computing library **Scipy** [58] (https://www.scipy.org/)
- Plotting library **Seaborn** (https://seaborn.pydata.org/)
- Machine learning library **Scikit-learn** [59] (https://scikit-learn.org/)
- Machine learning library **Sonnet** [60] (https://github.com/deepmind/sonnet/)
- Machine learning framework **TensorFlow** [61] (https://github.com/tensorflow/tensorflow/)
- Machine learning library **XGBoost** [62] (https://github.com/dmlc/xgboost/)

3 Procedure

The protocol consists of 34 steps, broken into the following stages: formal problem definition (6 steps); data pre-processing (10 steps); architecture selection (9 steps); risk calibration and uncertainty (4 steps); and model generalisability evaluation (5 steps).

Formal problem definition

1. **Evaluate the feasibility of a clinical use case:** For this protocol to be applicable to a particular clinical use case, it must meet several basic conditions: (i) there should be a computable definition of the target outcome or sufficiently large manually-labelled training dataset; (ii) there must be predictive signal in routinely collected structured EHR data (might it be possible for a specialist clinician to forecast this?); and (iii) the temporal granularity of the EHR must be compatible with the actionable time window of the prediction

(typically, the latter is a multiple of the former). Refine the research question through a cycle of clinician and patient engagement, referencing clinical guidelines and literature evidence as appropriate. Map out real world clinical pathways and identify opportunities for the output of the model to be integrated into existing workflows, including who the likely end users will be and the potential channels for model output (e.g. interruptive versus non-interruptive alerts [63]).

2. **Define outcome labels:** Identify an appropriate ground truth outcome label for supervised learning. One approach is to use e-alerting criteria, such as the NHS AKI e-alert [64] or the St Johns Sepsis Agent [65], which only use data available up to the time of the trigger. Another is to use data from the entire admission to timestamp outcomes of interest - such as using downstream outcomes to identify the most severe cases of AKI or sepsis. Outcomes may also be defined based on clinician actions (e.g. sepsis definitions based on the collection of a blood culture or antibiotic prescriptions) [66]; however this may introduce biases and encourages the model to predict outcomes only on those patients who have historically been investigated/treated. The gold standard is manual chart review, which may be used in conjunction with one or more of the above methods to validate the labelling approach. One major issue in EHR data is label leakage - where explicit indications of the outcome label are present at an earlier timepoint. Some studies suggest enforcing a gap time between the outcome label and the prediction trigger to reduce this risk [67]. In some cases, the width of the timesteps may have the effect of introducing a time buffer (although the gap is variable) - e.g. in [12], all entries in the same 6h bucket as the outcome label were excluded from the model input.

3. **Assess dataset quality:** Produce a formal dataset specification with descriptive statistics about each data element, including outliers and missingness. Assess the distributions of vitals and laboratory tests and compare against known physiological ranges. Harmonise admission records based on length of stay to concatenate overlapping admissions. As-

sess and report on the demographic diversity of the dataset. Consider using mitigation strategies to address data imbalance, for example oversampling/undersampling. Compile a random sample of the data for manual assessment by clinical experts, with particular focus on the fidelity of the outcome label.

4. **Define inclusion and exclusion criteria:** In support of a future prospective deployment, inclusion/exclusion criteria should be defined based on baseline criteria available from the point where model inference begins, rather than retrospective criteria such as percent missingness. Consider patient-level factors such as demographics, as well as environmental factors such as clinical setting or ward.

5. **Define time formulation:** Define the trigger time(s) and lookahead window(s) for prediction tasks (Figure 1). The trigger time refers to when, during an admission, inference will be performed - it may be a single static prediction e.g. at 24 hours after admission; continuous predictions e.g. triggered on an hourly basis; or dynamic predictions triggered when some criteria are satisfied e.g. when vitals are out of range. The lookahead window is the time interval after the trigger time in which the endpoints are defined. For AKI prediction we trialled lookahead windows ranging from 6 to 72 hours in 6h increments. The trigger time and lookahead window should be guided by domain knowledge about when early clinical markers may manifest and the window within which a prediction is clinically actionable. The interventions for AKI include medication review, fluid management, septic workup, etc., all of which may be effective in the 48h preceding AKI onset [68]. By contrast, for the mortality prediction task in Section 4 we also consider longer lookaheads (30, 90 days) which may be appropriate for sub-acute decisions around limits of care and palliative care referrals as per [19].

6. **Identify auxiliary prediction targets:** Auxiliary prediction targets can help to improve model performance on the primary prediction task, because concurrently learning multiple clinically-related endpoints may lead to a better internal data representation. The choice of

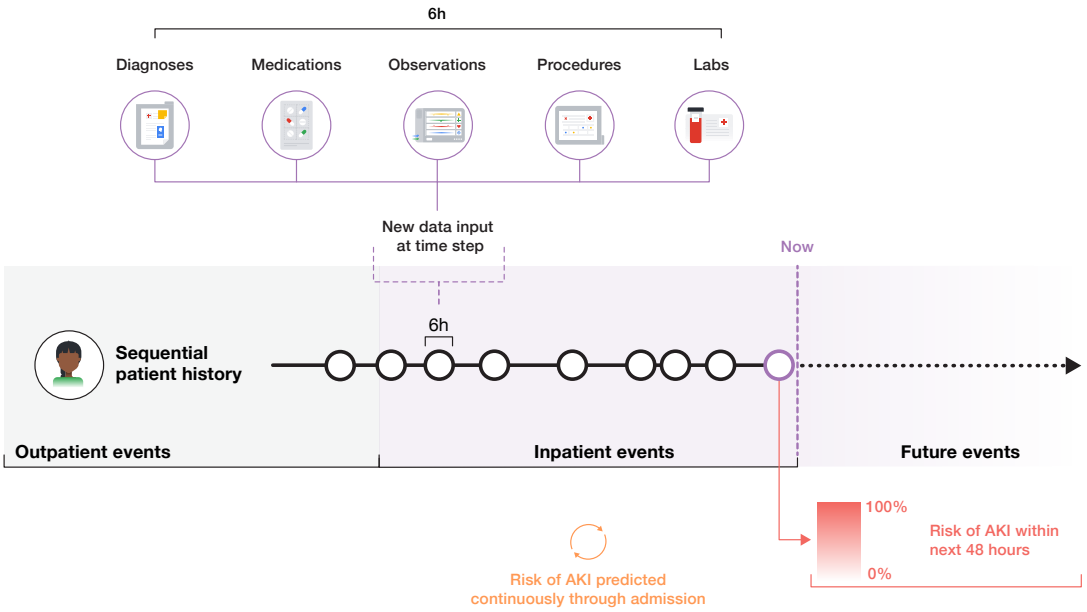

Figure 1 | Sequential risk prediction from EHR data: Rolling predictions using structured EHR data, here illustrating the prediction of AKI within 48h at a 6-hrly triggering frequency.

auxiliary tasks and how to configure the losses are topics of active research [20, 32, 69, 70]. At a high-level, the goal is to identify physiological observations directly related to the primary clinical endpoint. In the case of AKI, we used the maximum values (across the same set of lookahead windows as for the primary AKI endpoint) of 7 relevant laboratory tests known to rise in conjunction with AKI: creatinine, urea nitrogen, sodium, potassium, chloride, calcium and phosphate. Auxiliaries were modelled as separate *regression* tasks at each timestep, but the losses were combined into a single auxiliary loss. Auxiliary prediction tasks could potentially be used to model competing risks, as well as regularise the model and help with explainability of predictions. The panel of auxiliary tasks may be refined during training based on the performance uplift on the validation set.

Data pre-processing

7. **Create data splits:** Partition the dataset by randomly allocating patients into the following splits: *training*, *validation*, *calibration* and *test*. The entire patient record should be allocated to a single split, rather than having admissions separated across splits which risks information leakage. The minimum size of each split needs to be sufficient to derive valid statistical conclusions and should be based on an appropriate power calculation. For sufficiently large datasets, assigning 80% of the data to the training split, 5% to the validation split, 5% to the calibration split and 10% to the test split is a reasonable choice [12]. Only the training set is used for model development (Steps 8-25), with the test set held out from any analyses until the final model parameters are fixed. A calibration split is especially critical if the model is going to be used as a risk score or for continuous alerting (see Step 26).

8. **Feature engineering:** Seek input from clinicians and informaticians familiar with the source data to identify the most relevant subset of features from the complete input space. Features may be eliminated because the data quality is poor, they are clinically unrelated, too site-specific, etc. Identify a set of manually-engineered features that may hold predictive value. Examples include clinically-relevant ratios (e.g. ratio of blood urea nitrogen to serum creatinine) and interaction terms (see Supplementary Materials in [12]). These manually-engineered features are most important for the baseline models against which the deep models will be evaluated. Importantly, the feature engineering pipeline should be defined using the training set.

9. **Generate a sequential representation of patient data:** First, define the length of the discrete time window (timestep size). This was set as 6h in [12]; however can be on the order of minutes to days depending on the granularity of the data and triggering frequency of the prediction. Note that the timestep size must be less than or equal to the triggering frequency. There is a trade-off to be made in selecting the timestep size: short timesteps

risk being empty due to irregular sampling of EHR entries; however longer timesteps may introduce lossiness as the ordering of events is not preserved within each step. Repeated values for a given feature within a time bucket must be aggregated - typically using the mean or median, but other aggregation functions are valid. For entries where the timestamp is unknown, use a surrogate bucket - e.g. many EHR events may be associated with a day but no specific timestamp, so can be grouped into a surrogate bucket. This bucket is assigned to the end of that day to prevent leakage of future information in previous buckets. For empty timesteps during intervals where inference must be regularly triggered (e.g. during an inpatient admission), include an empty set. Finally, concatenate the entire patient record into a *sequential representation* running from the first to the last available data point, organised into distinct clinical events within inpatient and outpatient episodes (see Figure 2). Labels for the primary outcome and auxiliary targets should be appended at each timestep.

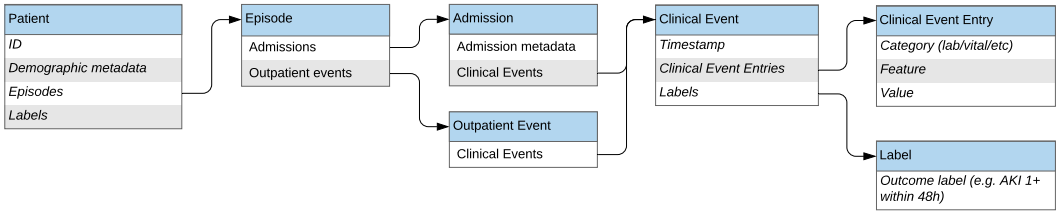

Figure 2 | Sequential representation: The sequential representation consists of a sequence of clinical event entries grouped into clinical events for each time bucket, which in turn are grouped by episode (admissions versus outpatient events).

10. **Clean EHR timestamps:** Beware of entries that may have a discrepancy between the EHR timestamp and the true availability of the data to clinicians. In the raw dataset used in [12], for example, diagnosis codes were uniformly timestamped at the beginning of admission even though the actual diagnosis was likely made at a different time (ICD codes being typically assigned at the time of discharge). Although this granularity was not available in the VA dataset, it may be possible to delineate several important timestamps for each entry - e.g. a laboratory test might have timestamps for when the order was

placed, when the sample was collected and when the result was visible in the EHR. It may be worth encoding each of these timesteps as distinct events or only using the lattermost timestamp. Where there is ambiguity around timestamps, move the relevant entries to the end of the relevant episode to avoid information leakage.

11. **Aggregate historical features:** For the baseline (non-recurrent) models that do not explicitly handle temporal input, generate a set of historical aggregate features. The lookback duration will depend on the clinical endpoint and should be guided by domain expertise. Define a set of statistical functions to use for feature aggregation, e.g. count, mean, median, standard deviation, minimum, maximum, average difference between subsequent measurements (these must be treated as distinct features, leading to a dimensionality increase). For non-numerical features, record a binary flag for whether they were present in the lookback window. In practice, we used 48 hours for shorter history, and longer historical trends were captured by considering 6 months, 1 year or 5 years prior.

12. **Vectorise the event sequence:** Vectorisation refers to transforming the sequential data representation into a feature vector appropriate for model input at each timestep. Since there can be valuable information in the pattern of missingness, a useful strategy for continuous features is to explicitly encode binary indicator variables (*presence features*) to enable the model to distinguish between a missing value in that timestep and a numerical value of zero [71]. Although zero imputation in conjunction with presence features was used in [12], there are numerous imputation strategies available for missing data including carry-forward, mean/median, and physiological reference imputation; as well as more advanced methods that have shown promise in EHR timeseries including multivariate imputation by chained equations (MICE) [72], Gaussian processes [73], generative adversarial networks [74], etc. Continuous features may also be associated with additional indicator variables denoting whether the value is high/normal/low based on local reference ranges. Represent categorical features using one-hot encoding.

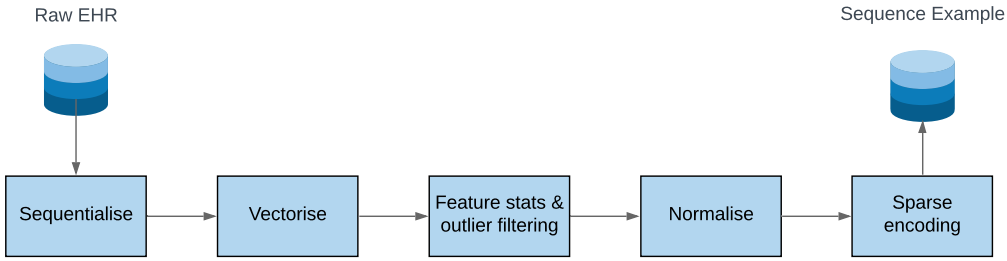

Figure 3 | **Pre-processing workflow:** Conversion of raw EHR data into a sequential representation, followed by vectorisation, normalisation (using feature statistics computed on the training data) and sparse encoding.

13. **Cap outlier numerical values:** For every input feature, cap values at the the 1st and 99th percentile values on the training split (or appropriate maximum/minimum bounds guided by clinician input). This is important as data entry errors can occur resulting in physiologically implausible, extreme outliers e.g. age above 150 years.

14. **Normalisation and sparse encoding:** To improve convergence speed [75], normalise the capped numerical input features to unit range or standardise to unit variance. Both approaches yielded similar results in [12]. Sparse encoding allows for a more efficient data representation of the sparse EHR feature space where only the explicitly non-zero values are represented. The sparse tensor consists of separate dense tensors denoting indices, feature values and original dense shape. This can then be converted to a required sequence example format for model input (Figure 3).

15. **Select performance metrics:** Define a set of relevant metrics for the primary use case as well as the auxiliary prediction targets. Select a) development metrics to be used for architecture selection; and b) final evaluation metrics to report. For classification tasks, use both the area under the precision-recall curve (PR AUC) and the area under the receiver operating characteristic curve (ROC AUC) in model development, since PR AUC is better suited to class imbalances [76]. Early prediction histograms are also valuable for fixed-window prediction tasks to demonstrate the latency between prediction and outcome (see

Figure 3 in [12]). Time-to-event or survival modelling is an alternate approach that may be used for continuous prediction tasks, for which there are emerging deep learning formulations [77]. For all evaluation metrics in continuous predictions tasks, there is an important distinction between timestep-level metrics and outcome-level metrics - e.g. the timestep precision and recall can be calculated by averaging the performance across all timesteps for which the model is being evaluated; however we can also calculate an outcome-level recall by examining what percentage of the outcomes (e.g. AKI episodes) have at least one correct prediction within a 48h window preceding onset. For both metrics, it may be acceptable depending on the clinical scenario to introduce tolerance in evaluation - e.g. accepting a positive prediction 48-60h prior to AKI onset as a true positive (as opposed to a false positive under a strict 48h lookahead). Results in [12] and Section 4 were computed without a tolerance buffer.

16. **Interval censoring:** Define interval censoring masks for both model training and evaluation. A mask refers to a sequence of binary flags overlaid on the event sequence, which indicates whether a timestep is included in training or evaluation. For example, in the AKI prediction use case, patients undergoing dialysis were excluded both from training and testing splits using a number of procedure codes to define the mask. Importantly, training and evaluation masks may be different. For example, intervals where the patient had AKI were included in the training procedure as there are still valuable physiological relationships between this timestep and future creatinine values; however they were excluded from evaluation as these timesteps would not be alerted on in practice. Adapting the training and evaluation masks can enable versatile experimental setups - e.g. predicting inpatient mortality only at points where the patient triggers a NEWS2 alert. After evaluation masks have been defined, it is possible to compute the outcome prevalence at an episode level (i.e. percentage of patients with AKI, or number of distinct AKI segments) and at a timestep level (i.e. percentage of timesteps with a positive label for AKI within 48h). This class distribution should be reported alongside a model to contextualise

the performance metrics.

Model architecture selection

17. **Train baseline models:** Train a panel of baseline models, such as logistic regression or XGBoost [62]. For these models, a subset of clinically-relevant and manually-engineered features (Step 8) may be selected. Interrogating the coefficients and feature importances of baseline models can assist in identifying label leakage, and guide redefinition of the outcome label if required. Confidence intervals for the performance metrics of baseline models should be calculated using a percentile bootstrap estimator [78].

18. **Feature embedding:** For each timestep, transform the sparse input tensors into a lower-dimensional continuous representation (i.e. embedding), that can be used as an input to the deep recurrent architecture. Multiply the sparse tensor by a lookup embedding matrix that is randomly initialized. If multiple features are present at a given timestep, aggregate the lookup embeddings - they were summed in [12] but the aggregation function can be tuned. Pass this to a multi-layer perceptron (MLP) embedding module with residual connections and L_1 regularisation to reduce overfitting. Sweep over a range of embedding sizes (a 2-layer embedding module with size 400 was used in [12]). In [12], there were separate embedding modules for different types of input features (numerical versus presence), and the outputs were then concatenated. Autoencoders (AEs) or variational autoencoders (VAEs) may be trialled, as these have shown promise in learning richer patient representations for predictive modelling [23, 79].

19. **Trial multiple deep architectures:** Implement a range of recurrent (and optionally convolutional) frameworks that receive the feature embeddings and feed into the primary and auxiliary output heads. Make the frameworks configurable with respect to recurrent cell types and their parameters, as well as different types of convolutional kernels. The following are some of the recurrent neural network (RNN) cells that can be trialled:

long short-term memory (LSTM) [80], update gate RNN (UGRNN), intersection RNN [81], simple recurrent unit (SRU) [82, 83], gated recurrent unit (GRU) [84], neural Turing machine (NTM) [85], memory-augmented neural network (MANN) [86], differentiable neural computer (DNC) [87], and relational memory core (RMC) [88]. Where there are multiple training heads (e.g. the primary output and the auxiliary tasks at various lookahead horizons), weights may be shared through the deep model, culminating in logistic layers specific to each task. Consider adding a cumulative distribution function to these logistic layers to encourage monotonicity in prediction outputs across overlapping lookahead horizons (e.g. risk of AKI within 48 hours should be greater than or equal to the risk within 24 hours). A comparison of deep architectures and baseline models is shown in Supplementary Table 4 in [12]. The architecture used in [12], shown in Figure 4, was a 3-layer LSTM [80] with highway connections [89] followed by linear layers for the primary outcome (AKI) and auxiliary heads (laboratory test regressions).

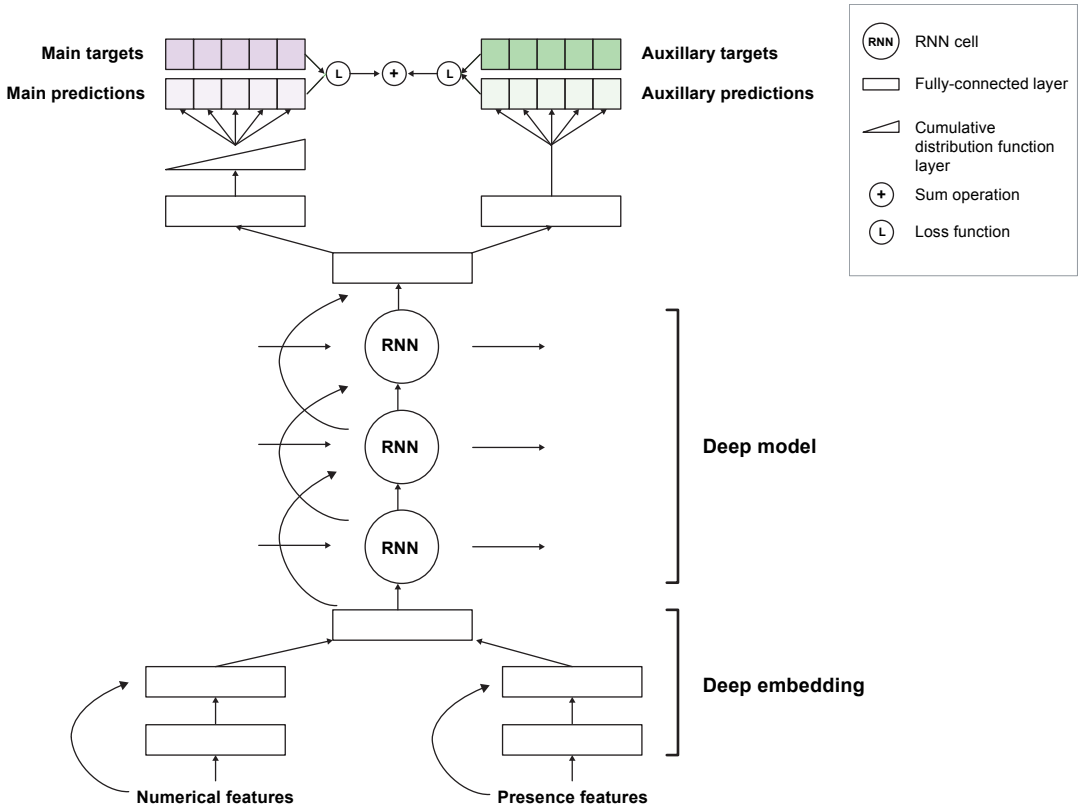

Figure 4 | Deep recurrent architecture: Numerical and presence features are embedded in parallel, feeding into a multi-task deep recurrent highway network architecture, with shared weights until final logistic layers for the primary versus auxiliary targets. The network was trained end-to-end including the embedding modules. Image source: [12].

20. **Set up the model optimiser:** By comparing the predicted output and the ground truth labels, compute a scalar loss value for each timestep. Next, compute scalar losses for each auxiliary task. In [12] the cross-entropy loss function (Bernoulli log-likelihood) was used for binary outcome prediction, and $L1/L2$ losses for the auxiliary laboratory test regressions. Optionally, re-weight the loss to account for skew in the target distribution. Define a composite loss as a weighted sum of primary and auxiliary losses, plus regularisation losses from the embedding module and model. Use the computed loss alongside a mini-batch optimization algorithm e.g. stochastic gradient descent, Adam, RMSProp etc. to iteratively adapt the weights of the neural network. Train using on the training data

split until convergence. Select the optimal learning regime (learning rate, decay) based on hyperparameter sweeps (See Step 21).

21. **Run an iterative sequence of hyperparameter sweeps:** Define hyperparameter sweeps based on domain knowledge and previous literature. Refer to Table 1 for the ranges of hyperparameters tested in [12]. Initially perform hyperparameter sweeps without auxiliary tasks to find a performant set of hyperparameters for the main task; then fine-tune the hyperparameters from that starting point, while executing sweeps to optimise the weight of the auxiliary loss. If target performance is not reached, revisit and expand earlier steps on data pre-processing and architecture selection. In each hyperparameter sweep, formulate a hypothesis for which architecture changes are likely to lead to performance improvements based on ML expertise. For example, overfitting is a significant risk, especially when the feature space is sparse and high dimensional. Mitigating strategies include reducing the RNN cell size, increasing regularisation, introducing drop-out or increasing the auxiliary loss weight. For each of the hyperparameter combinations, train a model on the training split and evaluate on the validation split (see Step 15), optimising for both PR AUC and ROC AUC. Select the best performing configuration at the end of this process as the final model architecture at this stage.

22. **Perform an ablation study:** The purpose of ablation is to minimise model complexity while preserving performance. Take the final model architecture from the previous step and define a set of components to remove (i.e. ablate) in order to assess their individual contributions. For example, this can include the number of stacked layers in the deep model, additional feature types like the historical aggregates, regularisation techniques, auxiliary prediction tasks, etc. For each ablation experiment, train a new model on the training set. Next, evaluate each of the ablated models on the validation set. To test for statistical significance, train a collection of ablation models, and calculate confidence intervals on the average performance using bootstrapping (see Step 28). If any of the

Table 1 | Hyperparameter sweeps used for the AKI model.

Hyperparameter	Values considered
RNN cell type	LSTM, GRU, UGRNN, SRU, Intersection RNN, MANN, NTM, DNC, RMC
RNN cell size	100, 150, 200, 250, 300, 400, 500
RNN num. layers	1, 2, 3
Embedding num. layers	1, 2, 3
Embedding dim. per feature type	200, 250, 300, 400, 500
Embedding combination	concatenate, sum
Embedding architecture type	MLP, AE, VAE
Embedding reconstruction loss weight	1e-2, 1e-3, 1e-4
Embedding reconstruction sampling ratio	1, 2, 5, 10
Optimise directly for PR AUC	on, off
Highway connections	on, off
Residual embedding connections	on, off
Input dropout	0, 0.1, 0.2, 0.3
Output dropout	0, 0.1, 0.2, 0.3
Embedding dropout	0, 0.1, 0.2, 0.3
Variational dropout	0, 0.1, 0.2, 0.3
Input regularisation type	None, L1, L2
Input regularisation term weight	1e-3, 1e-4, 1e-5
BPTT Window	32, 64, 128, 256, 512
Embedding activation functions	Tanh, ReLU [90], Leaky ReLu [91], Swish [92], ELU [93], SELU [94], ELiSH [95], Hard ELiSH [95], Sigmoid, Hard Sigmoid
Auxiliary task loss weight	0., 0.1, 0.5, 1, 5, 10
Learning rate	1e-2, 1e-3, 1e-4, 1e-5
Learning rate decay scheduling	on, off
Learning rate decay num. steps	6000, 8000, 12000, 15000, 20000
Learning rate decay base	0.7, 0.8, 0.85, 0.9, 0.95
Batch size	32, 64, 128, 256, 512
NTM/DNC memory capacity	64, 128, 256
NTM/DNC memory word size	16, 32, 64
NTM/DNC memory num. reads	6, 10
NTM/DNC memory num. writes	1, 2, 3

ablated models perform at least as well as the more complex final model, modify the architecture accordingly to favour the simpler version. Repeat this process until the model can no longer be simplified without significant loss of performance. Ablation results for AKI can be seen in Supplementary Tables 10 and 11 of [12].

23. **Compute feature saliency:** Estimating the contribution of individual features can aid in understanding what the model is learning, and can also be useful for quality assurance to protect against label leakage and spurious correlations. In this work, we performed occlusion analysis [96] which estimates a feature's contribution by evaluating the change in predicted risk when that feature is individually occluded. The occlusion process is

similar to replacing a feature by a *baseline* [97]. A feature was occluded by setting both its numerical value and associated presence feature to 0, i.e. we define the baseline as an *absent* feature. Feature attribution can then be evaluated by averaging the deviation in predicted risk under occlusion over an interval for a single patient (local saliency) or over all timesteps for the entire cohort (global saliency). Please note that feature saliency techniques were not tailored for multivariate heterogeneous time-series and we advise caution in their use.

24. **Failure case analysis:** Compute timestep-level and outcome-level metrics for all subjects in the validation set. Compile a set of representative success- and failure-cases (see Supplementary Material of [12] for example plots). Cases should be evaluated based on discriminative performance (was the ground-truth associated with this alert accurate?); as well as actionability (could this alert have impacted the clinical trajectory of this patient?). Detailed case review can be targeted towards certain clinical subgroups or cohorts where model performance is poor, in an attempt to tackle the pervasive issue of hidden stratification in model performance (where performance varies in clinically meaningful ways between patient subsets) [98].

25. **Define the final model architecture:** Define the resulting model architecture as final and do not revisit any of the previous steps at this point. Use the fixed set of parameters corresponding to this model to compute the predictions for all timesteps in all patients for each data split.

Risk calibration and uncertainty

26. **Calibration:** A well calibrated risk model is one where the predicted risk matches the incidence of the outcome of interest (i.e. 40% of patients with a 0.40 risk of AKI in 48h should develop an AKI in that timeframe). This is critical if a model is to be deployed as a clinical risk stratification tool. Deep learning models with softmax/sigmoid output trained

with cross-entropy loss are prone to miscalibration. Recalibration is often necessary to ensure that consistent probabilistic interpretations of the model predictions can be made [99]. Use the previously defined calibration set to align the predicted values with the underlying probability of the adverse event occurring at a given timestep. One approach is to fit an isotonic regression [100] model on the predictions against the target variable. Assess the quality of the calibration by comparing uncalibrated predictions to recalibrated ones using Brier score [101] and reliability plots [102] (see Extended Data Figure 3 in [12]).

27. **Estimate uncertainty of individual predictions:** To quantify the uncertainty of model predictions (i.e. prediction-level uncertainty), train an ensemble of multiple models with a fixed set of hyperparameters but different random initial seeds, similar to [103] and [104]. To get the uncertainty ranges for each prediction, take the set of predictions from all models and trim the distribution tails depending on the desired level of confidence. Note that alternative uncertainty estimation methods have been explored in the literature, including MC-dropout [105] and Bayesian neural networks [106] - the latter of which can enable efficient patient-level uncertainty estimation via a single model.

28. **Estimate performance uncertainty:** To gauge uncertainty on a trained model's performance (i.e. performance uncertainty), calculate confidence intervals of performance metrics (ROC AUC, PR AUC) using bootstrapping. First, sample patients from a single split with replacement (for 95% confidence intervals, take 200 records). Next, compute the pivot bootstrap estimator [107] using resampled values. Uncertainty estimates should be computed on the validation split during model development and on the test split for final performance metrics.

29. **Clinically-motivated operating points:** Performance metrics are dependent on the choice of an operating point (OP). Choose multiple OPs based on the PR curve of the final model and report the performance under each [108]. In consultation with clinical experts, evaluate which operating points are most clinically significant based on the val-

idation set metrics (e.g. for AKI an OP of two false positives for one true positive was chosen as being acceptable to assist a nephrology consult team in screening an inpatient population). Results for multiple OPs are shown in Figure 2 and Extended Data Table 4 in [12]. Note that since OPs are set on the validation split, they may not lie exactly on the PR curve reflecting test split performance.

Model generalisability evaluation

30. **Analyse model performance across subpopulations:** Define a set of clinical subpopulations relevant to the outcome of interest, which may include demographic and clinical characteristics. In [12] subgroups included patients with chronic kidney injury (CKD), diabetes and medical/surgical admissions. Report the performance including confidence intervals on each subpopulation. In particular, consider the performance and consequent resource allocation across protected groups (i.e. subpopulations vulnerable to health disparities) as part of a broader ML fairness evaluation [50], based on the available information in the research dataset.

31. **Quantify the expected daily alert rate:** Chronologically align all the patient timeseries from the test set. For each day in the longitudinal test set, compute the percentage of inpatients where the model: a) produced a true positive alert; b) produced a false positive alert without having provided a true positive alert within a certain prior time window; and c) did not produce any alerts. Compute the mean daily alert rate across all days in the longitudinal set. Report this metric to guide the likely resource burden in future prospective evaluation.

32. **Evaluate temporal generalisability on future unseen data:** Model performance may differ in important ways when prospectively deployed due to data drift [109]. To understand this potential risk steps 32 and 33 simulate the generalisability of models to future, unseen data and to previously unseen hospital sites. Choose a point in time t_P such that

approximately 80% of data entries occur prior to time t_P and approximately 20% occur after time t_P. Train a model using the final architecture determined in Step 25 using only data from prior to time t_P in the *training* split. Note that since hyperparameters were tuned using data from after t_P, this is an approximation and complete re-tuning with the pre-t_P test set would be the most rigorous approach. Generate model predictions for the entire *test* split. Generate 95% confidence intervals of PR AUC and ROC AUC for predictions made prior to t_P and for predictions made subsequent to t_P. Compare confidence intervals to determine if model performance on future unseen data is comparable to performance on historic data.

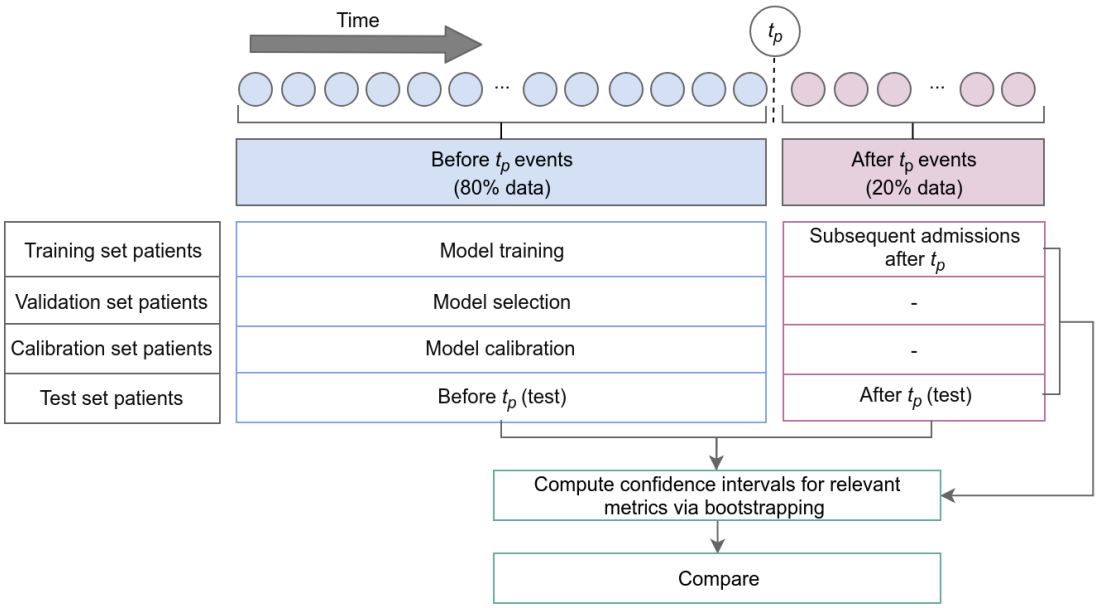

Figure 5 | Evaluating temporal generalisability on future unseen data. Confidence intervals for performance metrics on the *test* split prior to t_P are compared to those on the *test* split after t_P to determine if performance is preserved on future unseen data. Confidence intervals for performance metrics on the *train* split after t_P are compared to those on the *test* split after t_P to determine if there is a benefit from having prior historical data on patients present during model training.

33. **Evaluate regional generalisability in simulated cross-site deployments:** External validation, where model performance is computed on a different population/dataset, is a critical part of model evaluation. If multi-centre data are available, choose a split in hospital

sites such that approximately 80% of patient admissions occur at sites in group *A*, and approximately 20% occur at sites in group *B*. For single-site data, this split could be done in other ways, e.g. by ward. Train a model using the final architecture determined in Step 25 using the *training* split, excluding data entries from admissions at sites in group *B* (note the hyperparameter leakage issue in the step above). Run inference to generate model predictions for the *test* split. Compare performance metrics with confidence intervals to determine if model performance for unseen sites is comparable to performance for sites used during training.

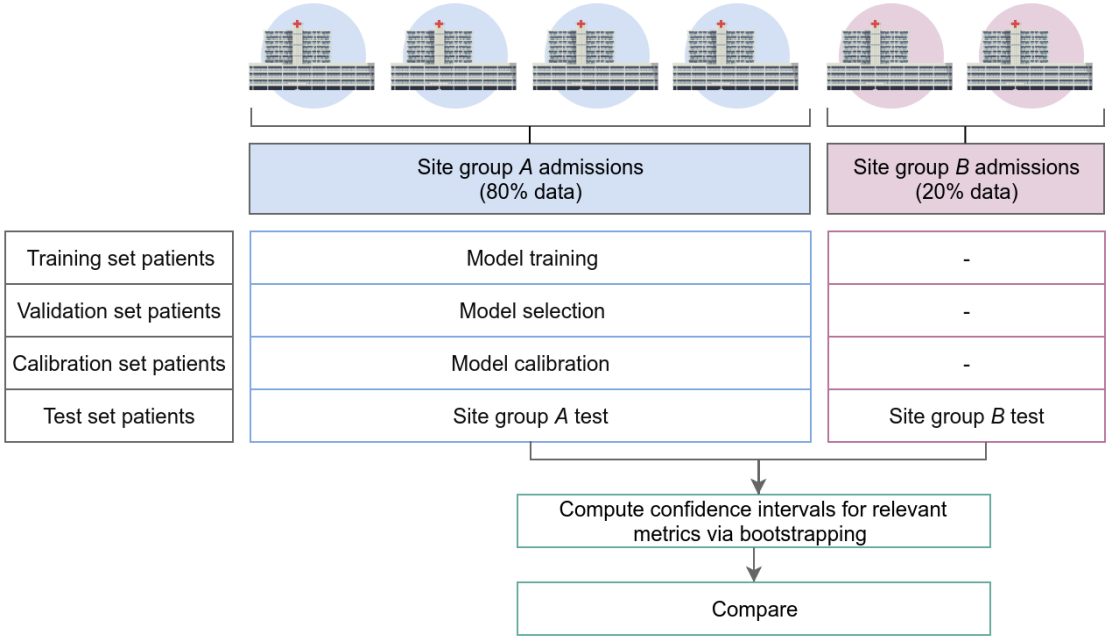

Figure 6 | Evaluating regional generalisability in simulated cross-site deployments: Confidence intervals for performance metrics on *test* split predictions made during admissions at sites in group *A* are compared to those for *test* split predictions made during admissions at sites in group *B* to determine if performance is preserved at sites that were unseen during training.

34. **Prospective evaluation:** There has been recent commentary around the 'last-mile' problem of machine learning implementation [36, 110]. Robust implementation research, consisting of a staged series of prospective observational and interventional studies, is a critical part of translating the above model into clinical use [35, 111]. Some of the goals of

implementation research include (i) defining the technical feasibility of data ingestion and inference; (ii) evaluating performance on prospective data and defining a model surveillance protocol to tackle issues such as drift; (iii) outcomes analysis on the clinical impacts of deployment under operational constraints (e.g. using decision curve analysis [112]). A detailed description of this prospective evaluation phase is outside the scope of this protocol.

3.1 Generalising protocol to new endpoints

In order to demonstrate that this protocol can be generalised to other tasks and time formulations, we describe below the key steps that were adapted to build predictive models for three other endpoints: mortality, length of stay and hospital readmission. These endpoints were chosen because the endpoint labels were possible to define in our dataset with reasonable fidelity; numerous ML performance benchmarks exist in the literature; and they are common operational focus areas where analytics may deliver value [47]. Furthermore, we demonstrate the performance of the architecture across a range of time formulations including varying the lookahead windows, trigger times and comparing continuous versus static prediction tasks. For each task, we also compare performance with and without auxiliary tasks.

Adjustments to Step 2 (outcome Labels):

- **Inpatient mortality:** The mortality label was based on a timestamped mortality flag, which included both in- and out-of-hospital mortality. To avoid label leakage, the sequential representation was masked from the 6h time bucket in which the mortality flag occurred. The inpatient mortality rate was 2.1%. For a more actionable model, ceilings of care should likely be factored into model design to avoid evaluation in situations where the patient has been identified as palliative by the treating team; however for the purposes of this methodological demonstration we do not explicitly exclude these subjects from evaluation.

- **Length of stay (LoS):** LoS was defined as the remaining length of stay from the trigger time. The median (interquartile range) LoS across all admissions was 3 (1-7) days with a mean of 9.7 days due to a number of very long inpatient admissions. Experiments were set up as binary classification tasks, predicting remaining LoS ≤ 2 or ≤ 7 days. Previous work has defined prolonged LoS prediction as *total* LoS above 7 days [14]; however for the purposes of showing multiple time formulations with a consistent lookahead window, we have chosen remaining LoS. To avoid label leakage, evaluation was not performed in the final time bucket of an admission. As Brock *et al.* suggest, LoS should likely be modelled using time-to-event analysis with mortality treated as a competing risk [113]. However, for the purposes of demonstrating architecture generalisability and comparing against literature benchmarks for LoS prediction, we model LoS independently here.

- **30-day readmission:** Readmission was defined as any inpatient admission to a VA facility within 30 days of hospital discharge. The percentage of discharged patients readmitted within this time window was 18.6%. Note that we do not factor in outpatient mortality events here.

Adjustments to Step 5 (time formulations):

To demonstrate versatility of the architecture to time formulations, the mortality and LoS tasks were set up as both continuous predictions (triggered every 6h) and static predictions (triggered at 24h or 48h after admission). Readmission was only modelled as a static task at the time of discharge. For static experiments, it is possible to train as a continuous task but only evaluate the model at a single time point (thereby converting it to a static task); however performance was found to be higher if both trained and evaluated statically. Regarding lookahead windows, mortality models were trained using the following intervals: 2, 7, 30, 90 days as well as a variable lookahead for in-hospital mortality. Remaining LoS was modelled based on 2 and 7 day cutoffs; readmission used a 30-day lookahead from the time of discharge.

Adjustments to Step 6 (auxiliary tasks):

For the mortality and LoS tasks, a panel of 14 laboratory tests was identified (extending on the 7 auxiliaries used for AKI, but within the scope of the available de-identified laboratory values): haemoglobin, white blood cell count, platelets, C reactive protein (CRP), international normalised ratio (INR), serum protein, albumin, glucose, creatinine, urea nitrogen, potassium, sodium, chloride and pH. We swept across multiple auxiliary configurations, varying the combination of aggregating functions (maximum, minimum, mean, standard deviation) and the combination of lookahead horizons (ranging from 6-72h). In all cases, auxiliary regressions were combined to give a single loss. Where a particular lab value was not measured, the loss was set to zero. For consistency, the setup of auxiliary tasks was kept constant for all time formulations of mortality and LoS, although auxiliary lookaheads could readily be customised. The intuition for keeping 48h was to capture the pattern of daily physiological trends for that patient even when modelling much longer term clinical outcomes (e.g. 30 day mortality). No auxiliaries were used for the 30-day readmission task as this was triggered only at the time of discharge.

Adjustments to Step 21 (hyperparameter sweeps):

Continuous mortality in admission without auxiliaries, static mortality without auxiliaries, and static mortality with auxiliaries used an initial learning rate of 0.0001. All other tasks used an initial learning rate of 0.001. Learning rate was decayed every 12,000 steps by a factor of 0.85, with batch size of 128 and back-propagation through time window of 128. Lookup embedding size varied from 200 to 400 depending on the task, with constant embedding layers of size 400 each for numerical and presence features. The RNN consisted of a 3-layer stacked LSTM with highway connections and cell size 300.

4 Anticipated results

Detailed results for various formulations of the AKI task are provided in [12]. Here we present new results for additional endpoints: mortality, length of stay and 30-day readmission (Tables 2 and 3).

For the continuous mortality prediction, PR AUC for the RNN with auxiliary tasks ranged from 38.3% for a 48 hour lookahead window to 73.8% for 90-day mortality, with ROC AUC of 98.6% and 95.6% respectively. These results compare favourably to literature benchmarks for mortality prediction, although performance comparisons are difficult across datasets and experimental formulations. A recent literature review of ML models in intensive care identified 70 papers predicting mortality [114]; however only a small subset of these used deep learning approaches and even fewer were designed for continuous predictions. Table A1 provides a summary of selected ML papers predicting inpatient mortality. Harutyunyan *et al.* [32] is one of the only studies to show results for continuous mortality predictions, specifically hourly prediction of mortality within 24 hours (which the authors refer to as 'physiologic decompensation'), showing PR AUC of 31.7% and ROC AUC of 90.5% on a dataset with significantly higher in-hospital mortality rate than the VA dataset (10.5% in their cohort from the Medical Information Mart for Intensive Care (MIMIC-III) dataset [115], versus 2.1% in the VA dataset). In a related experiment, Johnson *et al.* [116] simulated a real-time/continuous prediction task by training a gradient boosting model on MIMIC-III to predict in-hospital mortality at a random timepoint, with PR AUC 66.5% and ROC AUC 92.0%. More literature benchmarks exist for static formulations - most commonly, prediction of in-hospital mortality at 24h and 48h post admission. Our performance exceeds that reported on MIMIC-III structured data by [32, 67, 116]; however not the results reported by Puroshotham *et al.* on a feature set of 135 raw features using a multimodal RNN (PR AUC 78.6%, ROC AUC 94.1% for in-hospital mortality at 24h). A recent study by Brajer *et al.* prospectively and externally validated a model for predicting in-hospital mortality at the time of admission, with PR AUC and ROC AUC of 29%, 87% respectively on

retrospective validation and 14%, 86% on prospective validation [117].

Performance for remaining length of stay (PR AUC 93.3%, ROC AUC 84.3% for LoS <7 days at 24h post admission) and 30-day readmission (PR AUC 50.1%, ROC AUC 80.8%) also compare favourably to literature benchmarks - with Rajkomar *et al.* reporting ROC AUC of up to 86% for predicting total LoS >7 days at 24h post admission; and ROC AUC 77% for 30-day readmission by training over both structured data and notes [14]. Jamei *et al.* used an MLP to predict all-cause 30-day hospital readmission, and showed ROC AUC of 78% [118]; while Hilton *et al.* reported PR AUC and ROC AUC 38.3% and 75.8% respectively on 30-day readmission with a comparable outcome prevalence to our dataset of 14.2% [119].

We observe a modest performance uplift from the addition of auxiliary tasks, with static formulations for mortality and length of stay showing a 1-2% increase in mean PR AUC with preserved or increased ROC AUC. For continuous formulations, the performance uplift from auxiliaries was consistent but small (0-1% PR AUC gain) versus the 3.1% PR AUC uplift for the AKI task observed in [12]. It may be that a different panel of auxiliary endpoints, more closely tied to the primary outcome, would have led to a greater performance boost. Flexible approaches to automatically identify the optimal set of auxiliary tasks are beginning to emerge but are beyond the scope of this study [120].

Across all tasks and time formulations, the deep learning models trained using the above protocol outperformed baseline models (logistic regression and XGBoost). It has been suggested that performance on these canonical tasks saturates with simpler models [67]. These results suggest that there can still be significant gains in discriminative performance from deep architectures, however the marginal benefit may be higher for more complex clinical predictions.

More detailed results are provided for the model continuously predicting mortality in 48h. Figure 7 shows PR and ROC curves for the mortality in 48 hours model, annotated with multiple OPs (Step 29) for which the performance is further detailed in Table 4. At an operating point of 33% (one true positive to two false positives), 71.2% of deaths were predicted early within a window of up to 48h in advance (episode-level sensitivity). The shortcoming of episode-level

Table 2 | Continuous tasks: Model performance for continuous (i.e. regularly triggered) prediction tasks with variable lookahead windows. A comparison is made between two baseline models (logistic regression and XGBoost) and the deep recurrent architecture with and without auxiliary tasks. Outcome prevalence is the percentage of the positive class in the test set (timestep-level prevalence).

Task	Triggering	Timestep prevalence	Model	PR AUC [95% CI]	ROC AUC [95% CI]
Mortality in 48h	6hrly	0.42%	LR	11.2% [10.6, 11.8]	91.1% [90.7, 91.4]
			XGBoost	17.2% [16.2, 18.2]	94.1% [93.9, 94.3]
			RNN	37.4% [36.4, 38.3]	98.6% [98.5, 98.7]
			RNN with auxiliaries	38.3% [37.4, 39.5]	98.6% [98.6, 98.7]
Mortality in 7 days	6hrly	1.46%	LR	19.7% [18.9, 20.5]	89.5% [89.2, 89.9]
			XGBoost	25.9% [25.0, 26.9]	92.4% [92.1, 92.7]
			RNN	52.0% [51.1, 53.1]	97.9% [97.8, 98.0]
			RNN with auxiliaries	52.8% [51.8, 53.9]	98.0% [97.9, 98.1]
Mortality in 30 days	6hrly	4.7%	LR	32.5% [31.5, 33.5]	87.5% [87.1, 87.8]
			XGBoost	38.1% [37.1, 39.2]	90.0% [89.7, 90.7]
			RNN	66.8% [65.9, 67.9]	96.8% [96.6, 96.9]
			RNN with auxiliaries	67.7% [66.6, 68.8]	96.9% [96.7, 97.0]
Mortality in 90 days	6hrly	9.0%	LR	41.5% [40.6, 42.4]	85.9% [85.5, 86.4]
			XGBoost	47.1% [46.0, 48.2]	88.3% [88.0, 88.7]
			RNN	73.5% [72.6, 74.6]	95.5% [95.3, 95.7]
			RNN with auxiliaries	73.8% [72.6, 74.7]	95.6% [95.3, 95.8]
Mortality in admission	6hrly	4.9%	LR	27.5% [25.5, 29.1]	86.2% [85.2, 87.4]
			XGBoost	30.3% [28.2, 32.2]	87.3% [85.5, 89.1]
			RNN	63.5% [59.0, 67.0]	95.8% [94.8, 96.9]
			RNN with auxiliaries	64.5% [60.0, 68.8]	93.2% [89.7, 97.0]
Remaining LoS <= 2 days	6hrly	19.9%	LR	44.1% [43.8, 44.4]	78.7% [78.4, 79.0]
			XGBoost	50.5% [50.3, 50.8]	81.5% [81.2, 81.7]
			RNN	69.3% [69.1, 69.5]	90.0% [89.8, 90.1]
			RNN with auxiliaries	70.0% [69.8, 70.2]	90.2% [90.0, 90.3]
Remaining LoS <= 7 days	6hrly	40.7%	LR	73.4% [73.0, 73.7]	81.2% [80.9, 81.4]
			XGBoost	76.8% [76.5, 77.1]	83.2% [82.9, 83.4]
			RNN	86.2% [86.0, 86.4]	90.2% [90.0, 90.4]
			RNN with auxiliaries	86.4% [86.2, 86.6]	90.3% [90.1, 90.5]

sensitivity is that it does not account for the timeliness of predictions. To better visualize this, Figure 8 shows early prediction histograms for various OPs, demonstrating that performance is highest closest to the time of event.

Further research would be required to prepare this model for clinical deployment and evaluate it prospectively. Deep learning models for 1-year mortality have been used to guide palliative care referrals and end of life planning [19]. Acute mortality prediction may have utility in directing life-saving interventions, dovetailing with track-and-trigger systems such as NEWS2 and MEWS which have shown to improve outcomes when integrated with digital alerting systems [121]. Further prospective studies are required to evaluate whether ML models for acute

Table 3 | Static tasks: Model performance for static prediction tasks (i.e. triggered at a fixed point post admission). A comparison is made between two baseline models (logistic regression and XGBoost) and the deep recurrent architecture with and without auxiliary tasks. Outcome prevalence is the percentage of the positive class in the test set.

Task	Trigger time	Outcome prevalence	Model	PR AUC [95% CI]	ROC AUC [95% CI]
Mortality in admission	24h post admission	2.0%	LR	32.7% [31.4, 34.1]	94.1% [93.8, 94.3]
			XGBoost	40.8% [39.1, 42.5]	95.7% [95.5, 95.8]
			RNN	55.0% [53.4, 56.3]	97.6% [97.4, 97.7]
			RNN with auxiliaries	56.7% [55.3, 58.4]	97.8% [97.7, 98.0]
Mortality in admission	48h post admission	2.7%	LR	23.9% [22.6, 25.0]	88.1% [87.7, 88.5]
			XGBoost	31.1% [29.8, 32.5]	91.2% [90.9, 91.5]
			RNN	58.6% [56.9, 60.1]	97.2% [97.1, 97.4]
			RNN with auxiliaries	60.8% [59.2, 62.2]	97.5% [97.4, 97.7]
Remaining LoS <= 2 days	24h post admission	47.5%	LR	55.3% [54.9, 55.8]	69.1% [68.8, 69.4]
			XGBoost	59.5% [59.1, 59.9]	72.9% [72.6, 73.1]
			RNN	73.9% [73.5, 74.3]	82.0% [81.8, 82.2]
			RNN with auxiliaries	74.7% [74.4, 75.0]	82.6% [82.4, 82.7]
Remaining LoS <= 2 days	48h post admission	38.9%	LR	50.5% [50.0, 50.9]	67.2% [66.9, 67.5]
			XGBoost	55.9% [55.4, 56.4]	71.9% [71.6, 72.2]
			RNN	71.4% [70.9, 71.7]	81.3% [81.0, 81.5]
			RNN with auxiliaries	72.1% [71.7, 72.5]	81.9% [81.6, 82.1]
Remaining LoS <= 7 days	24h post admission	78.1%	LR	86.4% [86.1, 86.6]	72.2% [71.8, 72.5]
			XGBoost	88.4% [88.2, 88.6]	75.8% [75.6, 76.1]
			RNN	93.1% [92.9, 93.2]	83.9% [83.7, 84.1]
			RNN with auxiliaries	93.3% [93.2, 93.5]	84.3% [84.1, 84.5]
Remaining LoS <= 7 days	48h post admission	72.0%	LR	83.1% [82.8, 83.5]	70.8% [70.4, 71.2]
			XGBoost	85.7% [85.4, 86]	74.8% [74.5, 75.1]
			RNN	91.7% [91.5, 91.8]	83.4% [83.2, 83.7]
			RNN with auxiliaries	91.9% [91.7, 92.1]	83.8% [83.6, 84.1]
Readmission in 30 days	Discharge	18.7%	LR	30.2% [29.6, 30.7]	65.4% [65.0, 65.8]
			XGBoost	32.4% [31.8, 33.0	67.1% [66.8, 67.4]
			RNN	50.1% [49.1, 50.9]	80.8% [80.5, 81.1]

mortality prediction improve clinical outcomes when deployed, extending on prospective validation studies such as Brajer *et al.*[117]. It is important to note that mortality risk models may have unintended consequences if actively deployed in real-world settings, including pathological prediction cycles [122]. For example, a model may suggest removal of care for patients with a high mortality risk in turn providing confirmatory training data for the original risk. We emphasise that any live deployment of a mortality prediction model must undergo thorough ethical review and multiple stages of user-experience research and safety evaluation.

5 Conclusions

This protocol offers a versatile framework to develop deep learning prototypes for a range of clinical and operational use cases. The components of this protocol with greatest novelty include multi-tasking with physiological auxiliaries (Step 6); separate training and evaluation masks (Step 16); architecture ablation (Step 22); calibration and uncertainty estimation (Steps 26-28); clinically-motivated operating points (Step 29); and temporal/regional generalisability evaluations (Steps 32-33). It is critical that clinicians, informaticians and engineers are involved in an interdisciplinary team implementing this protocol. Further work is required in identifying appropriate clinical use cases and deploying the models in real-world settings via prospective implementation research. To support this next phase of translational research, the community must grapple with issues such as model safety, robustness and demographic biases in order to deliver meaningful clinical impact [40].

Table 4 | Operating points for mortality in 48h model: Percentage of mortality events detected up to 48h ahead of time at varying true positive (TP) to false positive (FP) operating points.

Precision	TP:FP ratio	Sensitivity
20%	1:4	84.2%
25%	1:3	79.1%
33%	1:2	71.2%
40%	2:3	62.4%
50%	1:1	51.7%
60%	3:2	40.7%
66%	2:1	35.1%
75%	3:1	28.2%

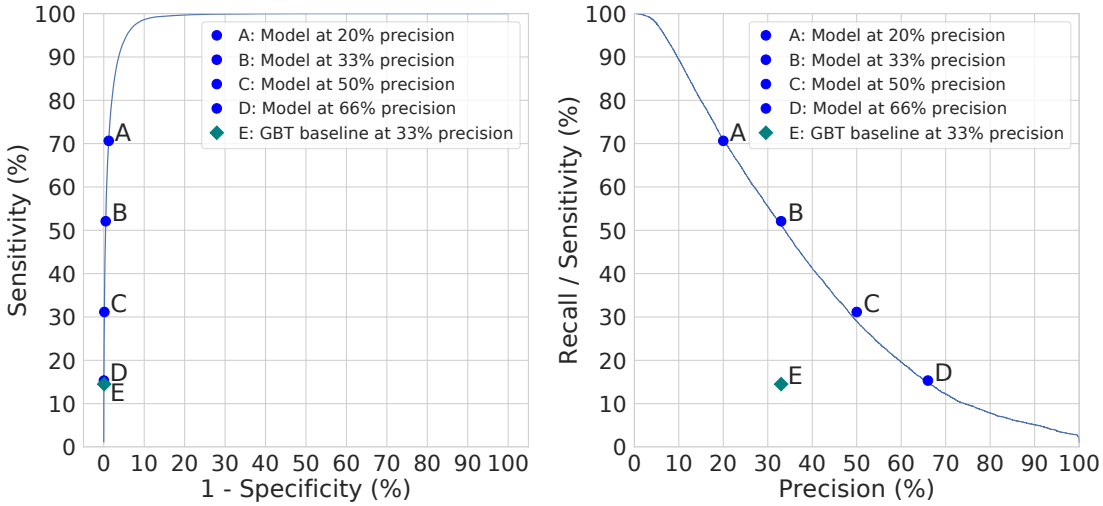

Figure 7 | Continuous prediction of mortality in 48h: Receiver operating characteristic (a) and precision–recall (b) curves for the risk of mortality in 48 hours. Blue dots represent different model operating points on the validation set.

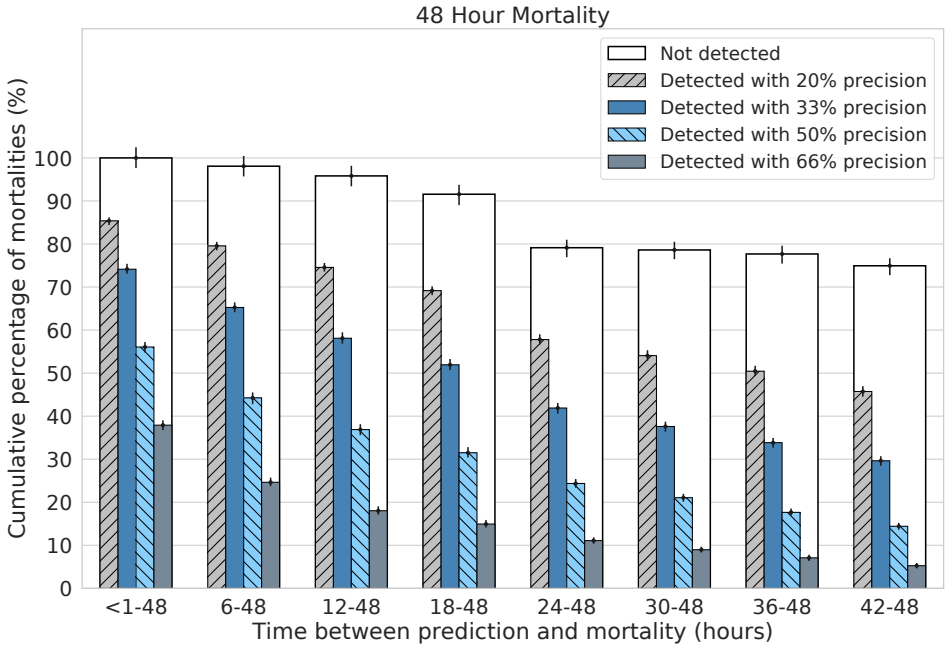

Figure 8 | Early prediction histogram for mortality in 48h: Model performance at timesteps prior to mortality. Error bars show bootstrap pivotal 95% confidence intervals; n = 200). The boxed area shows the upper limit on possible predictions for each time window.

Table A1 | Related works: Selected machine learning models for prediction of inpatient mortality. Where available, the inpatient mortality rate of the dataset is shown. Note MIMIC-III has different rates due to different subsets being used. Triggering tim refers to hours after admission when inference in triggered. 95% confidence intervals of performance are shown in brackets where available. GBM, Gradient BoostingMachine; GRU, Gated Recurrent Unit; GRU-D, Gated Recurrent Unit with Delay; FFNN, Feed Forward Neural Network; LSTM, Long Short Term Memory network; RNN, recurrent neural network.

Paper	Dataset (mortality rate)	No. of input features	Primary model	Triggering	Outcome	ROC AUC [95% CI]	PR AUC [95% CI]
Nakas et al. (2016) [123]	Inselspital Bern (2.4%)	23	Decision trees, FFNN	Static, at admission	In-hospital mortality	0.912	
Johnson et al. (2017) [116]	MIMIC-III	148	GBM	Static, 24h	In-hospital mortality	0.927	
				Random timepoint		0.920	0.665
Aczon et al. (2017) [124]	CHLA paediatric ICU (4.9%)	300	RNN (LSTM)	Static, 12h	In-hospital mortality	0.934	
Che et al. (2018) [125]	MIMIC-III (8.7%)	99	RNN (GRU-D)	Static, 48h	In-hospital mortality	0.853	
Purushotham et al. (2018) [49]	MIMIC-III (10.5%)	20	RNN (FFNN, GRU ensemble)	Static, 24h	In-hospital mortality	0.873	0.477
	MIMIC-III (10.5%)	135		Static, 48h		0.941	0.786
Rajkomar et al. (2018) [14]	Hospital A (2.1%)	Full FHIR embedding	RNN (ensemble LSTM)	Static, 24h	In-hospital mortality	0.95 [0.94-0.96]	
	Hospital B (2.5%)			Static, 24h		0.93 [0.92-0.94]	
Wang et al. (2019) [67]	MIMIC-III	103	RNN (GRU-D)	Static, 24h	In-hospital mortality	0.876	0.532
Caicedo-Torres et al. (2019) [126]	MIMIC-III (9.7%)	22	CNN	Static, 24h	In-hospital mortality	0.822	
				Static, 48h		0.874	
Mayampurath et al. (2019) [127]	U Chicago (2.5%)	156	CNN + recurrent layer	Static, 48h	In-hospital mortality	0.91 [0.90-0.92]	
Harutyunyan et al. (2019) [32]	MIMIC-III (13.2%)	17	RNN (multitask channel-wise LSTM)	Static, 24h	In-hospital mortality	0.870 [0.852-0.887]	0.533 [0.480-0.584]
				Continuous, 1-hourly	Mortality in 24h	0.905 [0.902-0.908]	0.317 [0.307-0.328]
Shickel et al. (2019) [34]	U Florida ICU (10.4%)	14	RNN (GRU)	Static, 24h	In-hospital mortality	0.89 [0.88-0.90]	
				Static, 48h		0.91 [0.90-0.91]	
	MIMIC-III (10.8%)	14		Static, 24h		0.90 [0.89-0.90]	
				Static, 48h		0.91 [0.91-0.92]	
Fritz et al. (2019) [128]	Barnes-Jewish intra-operative (1%)	56	Multi-path CNN	Randomly selected 1h interval	30-day mortality	0.867 [0.835–0.899]	0.095 [0.085-0.109]
Xia et al. (2019) [129]	MIMIC-III (11.7%)	50	RNN (ensemble LSTM)	Continuous, daily	28-day mortality	0.85	0.45
Nielsen et al. (2019) [130]	Danish ICU disease registry (33.4%)	44	FFNN	Static, 24h	In-hospital mortality	0.792	
Brajer et al. (2020) [117]	Duke (3.0%)	195	GBM	Static, at admission	In-hospital mortality	0.87 [0.83-0.89]	0.29 [0.25-0.37]
Hilton et al. (2020) [119]	Cleveland Clinic (1.4%)	171	GBM	Static, 24h	Mortality within 48-72h	0.91	

Competing financial interests

G.R., H.M. and C.L. are paid contractors of DeepMind/Google Health. The authors have no other competing interests to disclose.

Data and code availability

The clinical data used for the training, validation and test sets were collected at the US Department of Veterans Affairs and transferred to a secure data centre with strict access controls in de-identified format. Data were used with both local and national permissions. The dataset is not publicly available and restrictions apply to its use.

Code is available at GitHub link, illustrating the core components of the continuous prediction architecture, task configuration and auxiliary heads. The full data pre-processing pipeline is not included here as it is highly specific to this dataset. However, we do include synthetic examples of the pre-processing stages with an accompanying data-reading notebook. We believe this exemplar code can be appropriately customised to other EHR datasets and tasks.

Acknowledgements

We thank the veterans and their families under the care of the US Department of Veterans Affairs. We would also like to thank A. Graves, O. Vinyals, K. Kavukcuoglu, S. Chiappa, T. Lillicrap, R. Raine, P. Keane, A. Schlosberg, O. Ronneberger, J. De Fauw, K. Ruark, M. Jones, J. Quinn, D. Chou, C. Meaden, G. Screen, W. West, R. West, P. Sundberg and the Google AI team, J. Besley, M. Bawn, K. Ayoub and R. Ahmed. Special thanks to K. Peterson and the many other VA staff, including physicians, administrators and researchers who worked on the data collection. Thanks to the many DeepMind and Google Health colleagues for their support, ideas and encouragement.

G.R. & H.M. were supported by University College London and the National Institute for

Health Research (NIHR) University College London Hospitals Biomedical Research Centre. The views expressed are those of these author(s) and not necessarily those of the NHS, the NIHR or the Department of Health.

Author contributions

M.S., T.B., J.C., J.L., N.T., C.N., D.H. & R.R. initiated the project.

N.T., X.G., H.A., J.L., C.N. & C.B. created the dataset.

N.T., X.G., A.S., H.A., J.R., M.Z., A.M., I.P., N.H., S.B. & S.M. contributed to software engineering.

N.T., X.G., A.M., J.R., M.Z., A.S., S.M., N.H., S.B., M.G.S., X.G., J.L., T.O., C.N. & C.B. analysed the results.

N.T., X.G., A.M., J.R., M.Z., S.R. & S.M. designed the model architectures.

J.L., G.R., H.M., C.L., A.C., A.K., C.H., M.G.S, D.K., T.O. & C.N. contributed clinical expertise.

C.M., J.L., T.B., V.M., S.M. & C.N. managed the project.

N.T., J.L., M.G.S, J.R., M.Z., A.M., H.M., C.B., S.M. & G.R. wrote the paper.

Abbreviations

Abbreviation	Description
AE	Autoencoder
AKI	Acute Kidney Injury
AUC	Area Under Curve
CKD	Chronic Kidney Disease
CNN	Convolutional Neural Network
DNC	Differentiable Neural Computer
EHR	Electronic Health Record
FHIR	Fast Healthcare Interoperability Resources
FFNN	Feed Forward Neural Network
GBM	Gradient Boosting Machines
GRU	Gated Recurrent Unit
ICD-9	International Statistical Classification of Diseases and Related Health Problems
ICU	Intensive Care Unit
KDIGO	Kidney Disease: Improving Global Outcomes guidelines
LoS	Length of stay
LR	Logistic Regression
LSTM	Long Short-Term Memory Network
MANN	Memory-Augmented Neural Network
MLP	Multilayer Perceptron
NTM	Neural Turing Machine
PR	Precision/Recall
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
RMC	Relational Memory Core
ROC	Receiver Operating Characteristic
RRT	Renal Replacement Therapy
SRU	Simple Recurrent Unit
UGRNN	Update Gate Recurrent Neural Network
VA	US Department of Veterans Affairs
VAE	Variational Autoencoder

References

[1] "Acutely ill adults in hospital: recognising and responding to deterioration. nice guidance cg50," tech. rep., NICE: National Institute of Health and Care Excellence, United Kingdom, 2007.

[2] A. Rhodes, L. E. Evans, W. Alhazzani, M. M. Levy, M. Antonelli, R. Ferrer, A. Kumar, J. E. Sevransky, C. L. Sprung, M. E. Nunnally, B. Rochwerg, G. D. Rubenfeld, D. C. Angus, D. Annane, R. J. Beale, G. J. Bellinghan, G. R. Bernard, J.-D. Chiche, C. Coopersmith, D. P. De Backer, C. J. French, S. Fujishima, H. Gerlach, J. L. Hidalgo, S. M. Hollenberg, A. E. Jones, D. R. Karnad, R. M. Kleinpell, Y. Koh, T. C. Lisboa, F. R.

Machado, J. J. Marini, J. C. Marshall, J. E. Mazuski, L. A. McIntyre, A. S. McLean, S. Mehta, R. P. Moreno, J. Myburgh, P. Navalesi, O. Nishida, T. M. Osborn, A. Perner, C. M. Plunkett, M. Ranieri, C. A. Schorr, M. A. Seckel, C. W. Seymour, L. Shieh, K. A. Shukri, S. Q. Simpson, M. Singer, B. T. Thompson, S. R. Townsend, T. Van der Poll, J.-L. Vincent, W. J. Wiersinga, J. L. Zimmerman, and R. P. Dellinger, "Surviving sepsis campaign: International guidelines for management of sepsis and septic shock: 2016," *Critical Care Medicine*, vol. 45, no. 3, 2017.

[3] "Acute kidney injury: prevention, detection and management. nice guidance ng148," tech. rep., NICE: National Institute of Health and Care Excellence, United Kingdom, 2019.

[4] S. Suzuki, A. Yoshihisa, Y. Sato, Y. Kanno, S. Watanabe, S. Abe, T. Sato, M. Oikawa, A. Kobayashi, T. Yamaki, H. Kunii, K. Nakazato, T. Ishida, and Y. Takeishi, "Clinical significance of get with the guidelines–heart failure risk score in patients with chronic heart failure after hospitalization," *Journal of the American Heart Association*, vol. 7, no. 17, p. e008316, 2018.

[5] P. G. Lyons, D. P. Edelson, and M. M. Churpek, "Rapid response systems," *Resuscitation*, vol. 128, 2018.

[6] P. Yadav, L. Pruinelli, A. Hoff, M. Steinbach, B. L. Westra, V. Kumar, and G. J. Simon, "National early warning score (news) 2: Standardising the assessment of acute-illness severity in the nh," *CoRR*, vol. abs/1611.04660, 2016.

[7] C. P. Subbe, M. Kruger, P. Rutherford, and L. Gemmel, "Validation of a modified early warning score in medical admissions," *QJM : monthly journal of the Association of Physicians*, Oct 2001.

[8] C. van Walraven, I. A. Dhalla, C. Bell, E. Etchells, I. G. Stiell, K. Zarnke, P. C. Austin, and A. J. Forster, "Derivation and validation of an index to predict early death or unplanned

readmission after discharge from hospital to the community," *CMAJ : Canadian Medical Association journal = journal de l'Association medicale canadienne*, Apr 2010.

[9] J. Donzé, D. Aujesky, D. Williams, and J. L. Schnipper, "Potentially Avoidable 30-Day Hospital Readmissions in Medical Patients: Derivation and Validation of a Prediction Model," *JAMA Internal Medicine*, vol. 173, pp. 632–638, 04 2013.

[10] A. B. McCoy, L. R. Waitman, J. B. Lewis, J. A. Wright, D. P. Choma, R. A. Miller, and J. F. Peterson, "A framework for evaluating the appropriateness of clinical decision support alerts and responses," *Journal of the American Medical Informatics Association*, vol. 19, pp. 346–352, 08 2011.

[11] K. Kawamanto, M. C. Flynn, P. Kukhareva, D. ElHalta, R. Hess, T. Gregory, C. Walls, A. M. Wigren, D. Borbolla, B. E. Bray, M. H. Parsons, B. L. Clayson, M. S. Briley, C. H. Stipelman, D. Taylor, C. S. King, G. Del Fiol, T. J. Reese, C. R. Weir, T. Taft, and M. B. Strong, "A pragmatic guide to establishing clinical decision support governance and addressing decision support fatigue: a case study," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2018, pp. 624–633, Dec 2018.

[12] N. Tomasev, X. Glorot, J. W. Rae, M. Zielinski, H. Askham, A. Saraiva, A. Mottram, C. Meyer, S. Ravuri, I. Protsyuk, A. Connell, C. O. Hughes, A. Karthikesalingam, J. Cornebise, H. Montgomery, G. Rees, C. Laing, C. R. Baker, K. Peterson, R. Reeves, D. Hassabis, D. King, M. Suleyman, T. Back, C. Nielson, J. R. Ledsam, and S. Mohamed, "A clinically applicable approach to the continuous prediction of future acute kidney injury," *Nature*, 2019.

[13] K. E. Henry, D. N. Hager, P. J. Pronovost, and S. Saria, "A targeted real-time early warning score (trewscore) for septic shock," *Science Translational Medicine*, vol. 7, no. 299, pp. 299ra122–299ra122, 2015.

[14] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Mar-

cus, M. Sun, P. Sundberg, H. Yee, K. Zhang, Y. Zhang, G. Flores, G. E. Duggan, J. Irvine, Q. Le, K. Litsch, A. Mossin, J. Tansuwan, D. Wang, J. Wexler, J. Wilson, D. Ludwig, S. L. Volchenboum, K. Chou, M. Pearson, S. Madabushi, N. H. Shah, A. J. Butte, M. Howell, C. Cui, G. Corrado, and J. Dean, "Scalable and accurate deep learning with electronic health records," *NPJ Digital Medicine*, vol. 1, no. 1, 2018.

[15] J. L. Koyner, R. Adhikari, D. P. Edelson, and M. M. Churpek, "Development of a multicenter ward based AKI prediction model," *Clinical Journal of the American Society of Nephrology*, pp. 1935–1943, 2016.

[16] P. Cheng, L. R. Waitman, Y. Hu, and M. Liu, "Predicting inpatient acute kidney injury over different time horizons: How early and accurate?," in *AMIA Annual Symposium Proceedings*, vol. 2017, p. 565, American Medical Informatics Association, 2017.

[17] J. L. Koyner, K. A. Carey, D. P. Edelson, and M. M. Churpek, "The development of a machine learning inpatient acute kidney injury prediction model," *Critical Care Medicine*, vol. 46, no. 7, pp. 1070–1077, 2018.

[18] M. Komorowski, L. A. Celi, O. Badawi, A. Gordon, and A. Faisal, "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care," *Nature Medicine*, vol. 24, pp. 1716–1720, 2018.

[19] A. Avati, K. Jung, S. Harman, L. Downing, A. Ng, and N. H. Shah, "Improving palliative care with deep learning," *BMC medical informatics and decision making*, vol. 18, pp. 122–122, Dec 2018.

[20] B. Lim and M. van der Schaar, "Disease-Atlas: Navigating disease trajectories with deep learning," *Proceedings of Machine Learning Research*, vol. 85, 2018.

[21] J. Futoma, S. Hariharan, and K. A. Heller, "Learning to detect sepsis with a multitask gaussian process RNN classifier," in *Proceedings of the International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), pp. 1174–1182, 2017.

[22] P. Nguyen, T. Tran, N. Wickramasinghe, and S. Venkatesh, "Deepr: A convolutional net for medical records," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 22–30, 2017.

[23] R. Miotto, L. Li, B. Kidd, and J. T. Dudley, "Deep Patient: An unsupervised representation to predict the future of patients from the electronic health records," *Scientific Reports*, vol. 6, no. 26094, 2016.

[24] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," *International Conference on Learning Representations*, 2016.

[25] P. Z. J. H. Yu Cheng, Fei Wang, "Risk prediction with electronic health records a deep learning approach," in *Proceedings of the SIAM International Conference on Data Mining*, pp. 432–440, 2016.

[26] H. Soleimani, A. Subbaswamy, and S. Saria, "Treatment-response models for counterfactual reasoning with continuous-time, continuous-valued interventions," *arXiv Preprint arXiv:1704.02038*, 2017.

[27] A. M. Alaa, J. Yoon, S. Hu, and M. van der Schaar, "Personalized risk scoring for critical care patients using mixtures of gaussian process experts," *arXiv Preprint arXiv:1605.00959*, 2016.

[28] A. Perotte, N. Elhadad, J. S. Hirsch, R. Ranganath, and D. Blei, "Risk prediction for chronic kidney disease progression using heterogeneous electronic health record data and time series analysis," *Journal of the American Medical Informatics Association*, vol. 22, no. 4, pp. 872–880, 2015.

[29] A. Bihorac, T. Ozrazgat-Baslanti, A. Ebadi, A. Motaei, M. Madkour, P. M. Pardalos, G. Lipori, W. R. Hogan, P. A. Efron, F. Moore, *et al.*, "MySurgeryRisk: Development and validation of a machine-learning risk algorithm for major complications and death after surgery," *Annals of Surgery*, 2018.

[30] C. Xiao, E. Choi, and J. Sun, "Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review," *Journal of the American Medical Informatics Association : JAMIA*, vol. 25, pp. 1419–1428, Oct 2018.

[31] A. Ashfaq, A. Sant'Anna, M. Lingman, and S. Nowaczyk, "Readmission prediction using deep learning on electronic health records," *Journal of Biomedical Informatics*, vol. 97, p. 103256, 2019.

[32] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Scientific Data*, vol. 6, no. 1, p. 96, 2019.

[33] J. Fagerström, M. Bång, D. Wilhelms, and M. S. Chew, "Lisep lstm: A machine learning algorithm for early detection of septic shock," *Scientific Reports*, vol. 9, no. 1, p. 15132, 2019.

[34] B. Shickel, T. J. Loftus, L. Adhikari, T. Ozrazgat-Baslanti, A. Bihorac, and P. Rashidi, "DeepSOFA: a continuous acuity score for critically ill patients using clinically interpretable deep learning," *Scientific Reports*, vol. 9, no. 1, p. 1879, 2019.

[35] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King, "Key challenges for delivering clinical impact with artificial intelligence," *BMC Medicine*, vol. 17, no. 1, p. 195, 2019.

[36] M. G. Seneviratne, N. H. Shah, and L. Chu, "Bridging the implementation gap of machine learning in healthcare," *BMJ Innovations*, 2019.

[37] M. P. Sendak, W. Ratliff, D. Sarro, E. Alderton, J. Futoma, M. Gao, M. Nichols, M. Revoir, F. Yashar, C. Miller, K. Kester, S. Sandhu, K. Corey, N. Brajer, C. Tan, A. Lin, T. Brown, S. Engelbosch, K. Anstrom, M. Elish, K. Heller, R. Donohoe, J. Theiling, E. Poon, S. Balu, A. Bedoya, and C. OBrien, "Sepsis watch: A real-world integration of deep learning into routine clinical care (preprint)," *JMIR Medical Informatics*, June 2019.

[38] P.-H. C. Chen, Y. Liu, and L. Peng, "How to develop machine learning models for healthcare," *Nature Materials*, vol. 18, no. 5, pp. 410–414, 2019.

[39] Y. Liu, P.-H. C. Chen, J. Krause, and L. Peng, "How to Read Articles That Use Machine Learning: Users' Guides to the Medical Literature," *JAMA*, vol. 322, pp. 1806–1816, 11 2019.

[40] J. Wiens, S. Saria, M. Sendak, M. Ghassemi, V. X. Liu, F. Doshi-Velez, K. Jung, K. Heller, D. Kale, M. Saeed, P. N. Ossorio, S. Thadaney-Israni, and A. Goldenberg, "Do no harm: a roadmap for responsible machine learning for health care," *Nature Medicine*, vol. 25, no. 9, pp. 1337–1340, 2019.

[41] M. Ghassemi, T. Naumann, P. Schulam, A. L. Beam, I. Y. Chen, and R. Ranganath, "Practical guidance on artificial intelligence for health-care data," *The Lancet Digital Health*, vol. 1, pp. e157–e159, Aug 2019.

[42] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24–29, 2019.

[43] O. Gottesman, F. Johansson, M. Komorowski, A. Faisal, D. Sontag, F. Doshi-Velez, and L. A. Celi, "Guidelines for reinforcement learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 16–18, 2019.

[44] W. Luo, D. Phung, T. Tran, S. Gupta, S. Rana, C. Karmakar, A. Shilton, J. Yearwood, N. Dimitrova, T. B. Ho, *et al.*, "Guidelines for developing and reporting machine learning predictive models in biomedical research: a multidisciplinary view," *Journal of medical Internet research*, vol. 18, no. 12, p. e323, 2016.

[45] G. S. Collins and K. G. M. Moons, "Reporting of artificial intelligence prediction models," *The Lancet*, vol. 393, pp. 1577–1579, Apr 2019.

[46] Department of Veterans Affairs, "Veterans Health Administration: Providing health care for Veterans." https://www.va.gov/health/, 2018 (accessed November 9, 2018).

[47] D. W. Bates, S. Saria, L. Ohno-Machado, A. Shah, and G. Escobar, "Big data in health care: Using analytics to identify and manage high-risk and high-cost patients," *Health Affairs*, vol. 33, no. 7, pp. 1123–1131, 2014.

[48] E. W. Steyerberg and Y. Vergouwe, "Towards better clinical prediction models: seven steps for development and an abcd for validation," *European heart journal*, vol. 35, no. 29, pp. 1925–1931, 2014.

[49] "Benchmarking deep learning models on large healthcare datasets," *Journal of Biomedical Informatics*, vol. 83, pp. 112 – 134, 2018.

[50] A. Rajkomar, M. Hardt, M. D. Howell, G. Corrado, and M. H. Chin, "Ensuring fairness in machine learning to advance health equity," *Annals of internal medicine*, vol. 169, Dec 2018.

[51] M. A. Gianfrancesco, S. Tamang, J. Yazdany, and G. Schmajuk, "Potential biases in machine learning algorithms using electronic health record data," *JAMA internal medicine*, vol. 178, pp. 1544–1547, Nov 2018.

[52] T. B. Murdoch and A. S. Detsky vol. 309, pp. 1351–1352, 04 2013.

[53] J. Kemp, A. Rajkomar, and A. M. Dai, "Improved hierarchical patient classification with language model pretraining over clinical notes," 2019.

[54] P. Yadav, L. Pruinelli, A. Hoff, M. Steinbach, B. L. Westra, V. Kumar, and G. J. Simon, "Causal inference in observational data," *CoRR*, vol. abs/1611.04660, 2016.

[55] M. Oberst and D. A. Sontag, "Counterfactual off-policy evaluation with gumbel-max structural causal models," *CoRR*, vol. abs/1905.05824, 2019.

[56] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[57] T. Oliphant, "NumPy: A guide to NumPy." http://www.numpy.org/, 2019 (accessed June 10, 2019).

[58] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python." http://www.scipy.org/, 2019 (accessed June 10, 2019).

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[60] M. Reynolds, G. Barth-Maron, F. Besse, D. de Las Casas, A. Fidjeland, T. Green, A. Puigdomènech, S. Racanière, J. Rae, and F. Viola, "Open sourcing Sonnet - a new library for constructing neural networks." https://deepmind.com/blog/open-sourcing-sonnet/, 2017.

[61] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[62] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[63] S. Blecker, R. Pandya, S. Stork, D. Mann, G. Kuperman, D. Shelley, and J. S. Austrian, "Interruptive versus noninterruptive clinical decision support: Usability study," *JMIR human factors*, vol. 6, pp. e12469–e12469, Apr 2019.

[64] R. Hill and N. Selby, "Acute kidney injury warning algorithm best practice guidance," tech. rep., NHS England, 12 2014.

[65] R. C. Amland and K. E. Hahn-Cover, "Clinical decision support for early recognition of sepsis," *American journal of medical quality : the official journal of the American College of Medical Quality*, vol. 31, no. 2, pp. 103–110, 2016.

[66] M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, R. S. Hotchkiss, M. M. Levy, J. C. Marshall, G. S. Martin, S. M. Opal, G. D. Rubenfeld, T. van der Poll, J.-L. Vincent, and D. C. Angus, "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)," *JAMA*, vol. 315, pp. 801–810, 02 2016.

[67] S. Wang, M. B. A. McDermott, G. Chauhan, M. C. Hughes, T. Naumann, and M. Ghassemi, "Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii," 2019.

[68] A. Khwaja, "KDIGO clinical practice guidelines for acute kidney injury," *Nephron Clinical Practice*, vol. 120, no. 4, pp. c179–c184, 2012.

[69] D. Y. Ding, C. Simpson, S. Pfohl, D. C. Kale, K. Jung, and N. H. Shah, "The effectiveness of multitask learning for phenotyping with electronic health records data," *arXiv Preprint arXiv:1808.03331*, 2018.

[70] J. Wiens, J. Guttag, and E. Horvitz, "Patient risk stratification with time-varying parameters: A multitask learning approach," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2797–2819, 2016.

[71] Z. C. Lipton, D. C. Kale, and R. C. Wetzel, "Directly modeling missing data in sequences with rnns: Improved classification of clinical time series," *CoRR*, vol. abs/1606.04130, 2016.

[72] B. K. Beaulieu-Jones, D. R. Lavage, J. W. Snyder, J. H. Moore, S. A. Pendergrass, and C. R. Bauer, "Characterizing and managing missing structured data in electronic health records: Data analysis," *JMIR medical informatics*, vol. 6, Feb 2018.

[73] Y. Xue, D. Klabjan, and Y. Luo, "Mixture-based multiple imputation model for clinical data with a temporal dimension," 2019.

[74] J. Yoon, J. Jordon, and M. van der Schaar, "Gain: Missing data imputation using generative adversarial nets," *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[75] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, (London, UK, UK), pp. 9–50, Springer-Verlag, 1998.

[76] T. Saito and M. Rehmsmeier, "The precision recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLOS One*, vol. 10, no. 3, 2015.

[77] C. Lee, J. Yoon, and M. v. d. Schaar, "Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data," *IEEE Transactions on Biomedical Engineering*, vol. 67, pp. 122–133, Jan 2020.

[78] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman and Hall/CRC Press, 1 ed., 5 1994.

[79] N. Sadati, M. Z. Nezhad, R. B. Chinnam, and D. Zhu, "Representation learning with autoencoders for electronic health records: A comparative study," *arXiv*, 2019.

[80] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[81] J. Collins, J. Sohl-Dickstein, and D. Sussillo, "Capacity and learnability in recurrent neural networks," *International Conference on Learning Representations*, 2017.

[82] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks," *International Conference on Learning Representations*, 2017.

[83] T. Lei and Y. Zhang, "Training RNNs as fast as CNNs," *arXiv Preprint arXiv:1709.02755*, 2017.

[84] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv Preprint arXiv:1412.3555*, 2014.

[85] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv Preprint arXiv:1410.5401*, 2014.

[86] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proceedings of the International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), pp. 1842–1850, 2016.

[87] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[88] A. Santoro, R. Faulkner, D. Raposo, J. Rae, M. Chrzanowski, T. Weber, D. Wierstra, O. Vinyals, R. Pascanu, and T. Lillicrap, "Relational recurrent neural networks," *arXiv Preprint arXiv:1806.01822*, 2018.

[89] J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber, "Recurrent highway networks," in *Proceedings of the International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70, pp. 4189–4198, 2017.

[90] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15, pp. 315–323, PMLR, 2011.

[91] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 30, p. 3, 2013.

[92] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *International Conference on Learning Representations*, 2018.

[93] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *International Conference on Learning Representations*, 2016.

[94] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems* (I. Guyon, U. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, pp. 971–980, 2017.

[95] M. Basirat and P. M. Roth, "The quest for the golden activation function," *arXiv Preprint arXiv:1808.00783*, 2018.

[96] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *European Conference on Computer Vision*, 2014.

[97] M. Ancona, C. Öztireli, and M. H. Gross, "Explaining deep neural networks with a poly-

nomial time algorithm for shapley values approximation," *CoRR*, vol. abs/1903.10992, 2019.

[98] L. Oakden-Rayner, J. Dunnmon, G. Carneiro, and C. Ré, "Hidden stratification causes clinically meaningful failures in machine learning for medical imaging," 2019.

[99] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), pp. 1321–1330, 2017.

[100] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 694–699, ACM, 2002.

[101] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950.

[102] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the International Conference on Machine Learning* (L. D. Raedt and S. Wrobel, eds.), pp. 625–632, ACM, 2005.

[103] J. D. Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin, G. van den Driessche, B. Lakshminarayanan, C. Meyer, F. Mackinder, S. Bouton, K. W. Ayoub, R. Chopra, D. King, A. Karthikesalingam, C. O. Hughes, R. A. Raine, J. C. Hughes, D. A. Sim, C. A. Egan, A. Tufail, H. Montgomery, D. Hassabis, G. Rees, T. Back, P. T. Khaw, M. Suleyman, J. Cornebise, P. A. Keane, and O. Ronneberger, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature Medicine*, vol. 24, pp. 1342–1350, 2018.

[104] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing*

Systems (I. Guyon, U. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, pp. 6402–6413, 2017.

[105] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, pp. 1050–1059, 2016.

[106] M. W. Dusenberry, D. Tran, E. Choi, J. Kemp, J. Nixon, G. Jerfel, K. Heller, and A. M. Dai, "Analyzing the role of model uncertainty for electronic health records," 2019.

[107] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.

[108] S. Romero-Brufau, J. M. Huddleston, G. J. Escobar, and M. Liebow, "Why the c-statistic is not informative to evaluate early warning scores and what metrics to use," *Critical Care*, vol. 19, no. 1, p. 285, 2015.

[109] B. Nestor, M. B. A. McDermott, W. Boag, G. Berner, T. Naumann, M. C. Hughes, A. Goldenberg, and M. Ghassemi, "Feature robustness in non-stationary health records: Caveats to deployable model performance in common clinical machine learning tasks," 2019.

[110] E. Coiera, "The last mile: Where artificial intelligence meets reality," *J Med Internet Res*, vol. 21, p. e16323, Nov 2019.

[111] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," *Nat. Med.*, vol. 25, pp. 44–56, Jan 2019.

[112] N. H. Shah, A. Milstein, and S. C. Bagley, PhD, "Making Machine Learning Models Clinically Useful," *JAMA*, vol. 322, pp. 1351–1352, 10 2019.

[113] G. N. Brock, C. Barnes, J. A. Ramirez, and J. Myers, "How to handle mortality when investigating length of hospital stay and time to clinical stability," *BMC medical research methodology*, vol. 11, pp. 144–144, Oct 2011.

[114] D. Shillan, J. A. C. Sterne, A. Champneys, and B. Gibbison, "Use of machine learning to analyse routinely collected intensive care unit data: a systematic review," *Critical Care*, vol. 23, no. 1, p. 284, 2019.

[115] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, p. 160035, 2016.

[116] A. E. W. Johnson and R. G. Mark, "Real-time mortality prediction in the intensive care unit," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2017, pp. 994–1003, 2017.

[117] N. Brajer, B. Cozzi, M. Gao, M. Nichols, M. Revoir, S. Balu, J. Futoma, J. Bae, N. Setji, A. Hernandez, and M. Sendak, "Prospective and External Evaluation of a Machine Learning Model to Predict In-Hospital Mortality of Adults at Time of Admission," *JAMA Network Open*, vol. 3, pp. e1920733–e1920733, 02 2020.

[118] M. Jamei, A. Nisnevich, E. Wetchler, S. Sudat, and E. Liu, "Predicting all-cause risk of 30-day hospital readmission using artificial neural networks," *PloS one*, vol. 12, pp. e0181173–e0181173, Jul 2017.

[119] C. B. Hilton, A. Milinovich, C. Felix, N. Vakharia, T. Crone, C. Donovan, A. Proctor, and A. Nazha, "Personalized predictions of patient outcomes during and after hospitalization using artificial intelligence," *npj Digital Medicine*, vol. 3, p. 51, Apr 2020.

[120] S. Liu, A. J. Davison, and E. Johns, "Self-supervised generalisation with meta auxiliary learning," *CoRR*, vol. abs/1901.08933, 2019.

[121] C. P. Subbe, B. Duller, and R. Bellomo, "Effect of an automated notification system for deteriorating ward patients on clinical outcomes," *Critical Care*, vol. 21, no. 1, p. 52, 2017.

[122] M. Ghassemi, T. Naumann, P. Schulam, A. L. Beam, I. Y. Chen, and R. Ranganath, "A review of challenges and opportunities in machine learning for health," 2018.

[123] C. T. Nakas, N. Schütz, M. Werners, and A. B. Leichtle, "Accuracy and calibration of computational approaches for inpatient mortality predictive modeling," *PLOS ONE*, vol. 11, pp. 1–11, 07 2016.

[124] M. Aczon, D. Ledbetter, L. V. Ho, A. M. Gunny, A. Flynn, J. Williams, and R. C. Wetzel, "Dynamic mortality risk predictions in pediatric critical care using recurrent neural networks," *CoRR*, vol. abs/1701.06675, 2017.

[125] Z. Che, S. Purushotham, K. Cho, and D. Sontag, "Recurrent neural networks for multivariate time series with missing values," *Scientific Reports*, vol. 8, no. 1, p. 6085, 2018.

[126] W. Caicedo-Torres and J. Gutierrez, "Iseeu: Visually interpretable deep learning for mortality prediction inside the icu," *Journal of Biomedical Informatics*, vol. 98, p. 103269, 2019.

[127] A. Mayampurath, L. N. Sanchez-Pinto, K. A. Carey, L.-R. Venable, and M. Churpek, "Combining patient visual timelines with deep learning to predict mortality," *PloS one*, vol. 14, pp. e0220640–e0220640, Jul 2019.

[128] "Deep-learning model for predicting 30-day postoperative mortality," *British Journal of Anaesthesia*, vol. 123, no. 5, pp. 688 – 695, 2019.

[129] J. Xia, S. Pan, M. Zhu, G. Cai, M. Yan, Q. Su, J. Yan, and G. Ning, "A long short-term memory ensemble approach for improving the outcome prediction in intensive care unit," *Computational and mathematical methods in medicine*, vol. 2019, pp. 8152713–8152713, Nov 2019. PMC6885179[pmcid].

[130] A. B. Nielsen, H.-C. Thorsen-Meyer, K. Belling, A. P. Nielsen, C. E. Thomas, P. J. Chmura, M. Lademann, P. L. Moseley, M. Heimann, L. Dybdahl, L. Spangsege,

P. Hulsen, A. Perner, and S. Brunak, “Survival prediction in intensive-care units based on aggregation of long-term disease history and acute physiology: a retrospective study of the danish national patient registry and electronic patient records,” *The Lancet Digital Health*, vol. 1, pp. e78–e89, Jun 2019.