

Temporal Pointwise Convolutional Networks for Length of Stay Prediction in the Intensive Care Unit

Emma Rocheteau
University of Cambridge
Cambridge, UK
ecr38@cam.ac.uk

Pietro Liò
University of Cambridge
Cambridge, UK
pl219@cam.ac.uk

Stephanie Hyland
Microsoft Research
Cambridge, UK
stephanie.hyland@microsoft.com

ABSTRACT

The pressure of ever-increasing patient demand and budget restrictions make hospital bed management a daily challenge for clinical staff. Most critical is the efficient allocation of resource-heavy Intensive Care Unit (ICU) beds to the patients who need life support. Central to solving this problem is knowing for how long the current set of ICU patients are likely to stay in the unit. In this work, we propose a new deep learning model based on the combination of temporal convolution and pointwise (1x1) convolution, to solve the length of stay prediction task on the eICU and MIMIC-IV critical care datasets. The model – which we refer to as Temporal Pointwise Convolution (TPC) – is specifically designed to mitigate common challenges with Electronic Health Records, such as skewness, irregular sampling and missing data. In doing so, we have achieved significant performance benefits of 18-68% (metric and dataset dependent) over the commonly used Long-Short Term Memory (LSTM) network, and the multi-head self-attention network known as the Transformer. By adding mortality prediction as a side-task, we can improve performance further still, resulting in a mean absolute deviation of 1.55 days (eICU) and 2.28 days (MIMIC-IV) on predicting remaining length of stay.

CCS CONCEPTS

• **Applied computing** → **Health informatics**; • **Computing methodologies** → **Neural networks**; **Multi-task learning**; • **Mathematics of computing** → **Time series analysis**.

KEYWORDS

Patient Outcome Prediction, Length of Stay, Mortality, Intensive Care Unit, Temporal Convolution

ACM Reference Format:

Emma Rocheteau, Pietro Liò, and Stephanie Hyland. 2021. Temporal Pointwise Convolutional Networks for Length of Stay Prediction in the Intensive Care Unit. In *ACM Conference on Health, Inference, and Learning (ACM CHIL '21)*, April 8–10, 2021, Virtual Event, USA. ACM, New York, NY, USA, 19 pages. <https://doi.org/10.1145/3450439.3451860>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ACM CHIL '21, April 8–10, 2021, Virtual Event, USA
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8359-2/21/04.
<https://doi.org/10.1145/3450439.3451860>

1 INTRODUCTION

In-patient length of stay (LoS) explains approximately 85-90% of inter-patient variation in hospital costs in the United States [44]. Extended length of stay is associated with increased risk of contracting hospital acquired infections [19] and mortality [28]. Hospital bed planning can help to mitigate these risks and improve patient experiences [1]. This is particularly important in the intensive care unit (ICU), which has the highest operational costs in the hospital [8] and a limited supply of specialist staff and resources.

At present, discharge date estimates are done manually by clinicians, but these rapidly become out-of-date and can be unreliable (for example Mak et al. [31] found that the average error made by clinicians was 3.82 days). Automated systems drawing on the electronic health record (EHR) have the potential to improve forecasting accuracy using state-of-the-art models that can be updated in light of new data. This has efficiency benefits in reducing the administrative burden on clinicians, and the improved accuracy may enable more sophisticated planning strategies e.g. scheduling high-risk elective surgeries on days with more availability [13].

In our work, we simulate real-time predictions in retrospective data by updating the patients' remaining ICU length of stay prediction at hourly intervals during their stay using the preceding data from the EHR (similar to Harutyunyan et al. [18]). When designing both the architecture and pre-processing, we focus on mitigating the effects of non-random missingness due to irregular sampling, sparsity, outliers, skew, and other common biases in EHR data. Our key contributions are:

- (1) A new model – Temporal Pointwise Convolution (TPC) – which combines:
 - Temporal convolutional layers [25, 61], which capture causal dependencies across the time domain.
 - Pointwise convolutional layers [29], which compute higher level features from interactions in the feature domain.
 Our model significantly outperforms the commonly used Long-Short Term Memory (LSTM) network [21] and the Transformer [62] by margins of 18-68%.
- (2) We make a case for using the mean-squared logarithmic error (MSLE) loss function to train LoS models, as it deals more naturally with positively-skewed labels.
- (3) By adding in-hospital mortality as a side-task, we demonstrate further performance gains in the multitask setting.
- (4) We perform several investigations to improve our understanding of the model, including: an extensive ablation study of the model architecture, a post-hoc analysis of feature importances with integrated gradients [56], and a visualisation to show the model reliability as a function of the time since admission and the predicted remaining LoS.

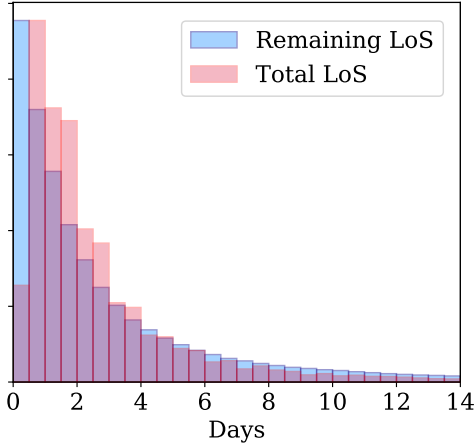


Figure 1: Total and remaining LoS distributions in the eICU dataset. The remaining LoS has a significant positive skew, with mean and median values of 3.47 and 1.67 days respectively. The skew in MIMIC-IV (not shown) is even more pronounced (5.70 and 2.70 days).

Additionally, we develop a data processing pipeline for the eICU [39] and MIMIC-IV [24] databases that is designed to i) mitigate some of the impact of sparsity (for the diagnoses) and missing data (for time series) in the EHR and ii) extract a wide variety of features semi-automatically such that the approach is generalisable to other EHR databases. Our code is available at: <https://github.com/EmmaRocheteau/TPC-LoS-prediction>.

2 RELATED WORK

Despite its importance, LoS prediction has received less attention than mortality prediction. This could be due to its difficulty; LoS depends heavily on operational factors and there is considerable positive skew in its distribution (see Figure 1). While it has been addressed as a regression problem (optimised using the mean-squared error (MSE) [42, 51]), it is often simplified into binary classification (short vs. long stay) [15, 35, 43], or as a multi-class task [18]. This simplification comes at a cost of utility, so we choose to focus on the more challenging regression variant.

Owing to the centrality of time series in the EHR, LSTMs have been by far the most popular model for predicting LoS [18, 43, 51]. This reflects the prominence of LSTMs in other clinical prediction tasks such as predicting in-hospital adverse events including cardiac arrest [60] and acute kidney injury [59], forecasting diagnoses, medications and interventions [5, 30, 57], missing-data imputation [3], and mortality prediction [4, 18, 52]. More recently, the Transformer model [62] been shown to marginally outperform the LSTM on LoS [53] (and it continues to dominate in many other domains [34]). Therefore, the LSTM and the Transformer were chosen as key baselines.

Temporal convolution models have previously been applied to the task of early disease detection using longitudinal lab tests [37, 45, 46], yielding similar results to the LSTM. We highlight two main differences in our work: we introduce a set of pointwise convolutions in parallel, and the temporal convolution filters do

not share their parameters between features, allowing the model to optimise processing in spite of heterogeneity in the temporal characteristics. We demonstrate via ablation studies how these design choices contribute substantial improvements to the patient state representation, yielding state-of-the-art results on LoS prediction.

3 METHODS

3.1 Model Overview

We want our model to extract both temporal trends and inter-feature relationships in order to capture the patient’s clinical state. Consider a patient who is experiencing slowly worsening respiratory symptoms but is otherwise stable. As this patient is unlikely to be weaned from their ventilator in the near future, a clinician might anticipate a long remaining LoS, but how do they come to this conclusion? Intuitively, one of the factors they are evaluating is the trajectory of the patient e.g. they may ask themselves “Is the respiratory rate getting better or deteriorating?”. However, they can obtain a better indication of lung function by combining certain features e.g. the $\text{PaO}_2/\text{FiO}_2$ ratio, and then looking at how *these* vary over time. A model should therefore be adept at extracting and combining both intra-feature temporal statistics and inter-feature relationships.

Formally, our task is to predict the remaining LoS at regular timepoints $y_1, \dots, y_T \in \mathbb{R}_{>0}$ in the patient’s ICU stay, up to the discharge time T , using the diagnoses ($\mathbf{d} \in \mathbb{R}^{D \times 1}$), static features ($\mathbf{s} \in \mathbb{R}^{S \times 1}$), and time series ($\mathbf{x}_1, \dots, \mathbf{x}_T \in \mathbb{R}^{F \times 2}$). Initially, for every timepoint t , there are two ‘channels’ per time series feature: F feature values ($\mathbf{x}'_t \in \mathbb{R}^{F \times 1}$), and their corresponding decay indicators ($\mathbf{x}''_t \in \mathbb{R}^{F \times 1}$). The decay indicators tell the model how recently the observation \mathbf{x}'_t was recorded. They are described in detail in Section 4. As we pass through the layers of our model, we repeatedly extract trends and inter-feature relationships using a novel combination of techniques.

3.2 Temporal Convolution

Temporal Convolution Networks (TCNs) [25, 61] are a subclass of convolutional neural networks [12] that convolve over the time dimension. They operate on two key principles: the output is the same length as the input, and there can be no leakage of data from the future. We use stacked TCNs to extract *temporal trends* in our data. Unlike most implementations including [45], we *do not share weights across features* i.e. weight sharing is only across time (like in Xception [6]). This is because our features differ sufficiently in their temporal characteristics to warrant specialised processing.

We define the temporal convolution operation for the i_{th} feature in the n_{th} layer as

$$(f^{n,i} * \mathbf{h}^{n,i})(t) = \sum_{j=1}^k f^{n,i}[j] \mathbf{h}_{t-d(j-1)}^{n,i} \quad (1)$$

where $\mathbf{h}_{1:t}^{n,i} \in \mathbb{R}^{C^n \times t}$ represents the temporal input to layer n up to timepoint t , which contains C^n channels per feature¹. The convolutional filter $f^{n,i} : \{1, \dots, k\} \rightarrow \mathbb{R}^{Y \times C^n}$ is a tensor of $Y \times C^n \times k$ parameters per feature. It maps C^n input channels into Y output

¹In the first layer, the input $\mathbf{h}_{1:t}^{n,i}$ is the original data $\mathbf{x}_{1:t}^{n,i} \in \mathbb{R}^{2 \times t}$, so $C^1 = 2$.

channels while examining k timesteps. The output is therefore $(f^{n,i} * h^{n,i})(t)^\top \in \mathbb{R}^{1 \times Y}$. The dilation factor, d , and kernel size, k , together determine the temporal receptive field or ‘timespan’ of the filter: $d(k-1) + 1$ hours for a single layer. To ensure that the output is always length T , we add left-sided padding of size $d(k-1)$ before every temporal convolution (not shown in equation 1). The $t - d(j-1)$ term ensures that we only look backwards in time. The receptive field can be increased by stacking multiple TCNs (as in Wavenet [61] and ByteNet [25]). We increment the dilation by 1 with each layer i.e. $d = n$.

We concatenate the temporal convolution outputs for each feature, i as follows

$$\underbrace{(f^n * h^n)}_{\text{Temp. Out. (2)}}(t) = \left\| \underbrace{(f^{n,i} * h^{n,i})(t)^\top}_{\text{Temp. In. (1)}} \right\|_{i=1}^{R^n} \quad (2)$$

We use $\|$ to denote concatenation i.e. $\|_{i=1}^A \mathbf{a}^i = \mathbf{a}^1 \parallel \dots \parallel \mathbf{a}^A$. In our case, the output dimensions are $R^n \times Y$, where R^n is the number of temporal input features. Throughout this section we label terms with numbers (1), (2) etc. corresponding to objects in Figure 3. We recommend following this alongside the equations.

3.3 Pointwise Convolution

Pointwise convolution [29], also referred to as 1×1 convolution, is typically used to reduce the channel dimension when processing images [58]. It can be conceptualised as a fully connected layer, applied separately to each timepoint (shown diagrammatically in Figure 2). As in temporal convolution, the weights are shared across all timepoints; however, there is no *information transfer* across time. Instead, information is shared across the *features* to obtain Z interaction features², $\mathbf{p}_t^n = (\mathbf{b}(\mathbf{h}_t^n) \parallel \mathbf{s} \parallel \mathbf{x}_t^n) \in \mathbb{R}^{P^n \times 1}$, where $P^n = (R^n \times C^n) + F + S$, and $\mathbf{b} : A^{d_1 \times d_2 \times \dots \times d_n} \rightarrow A^{(d_1 \cdot d_2 \cdot \dots \cdot d_n) \times 1}$ is the flatten operation. We define the pointwise convolution operation in the n_{th} layer as

$$\underbrace{(g^n * \mathbf{p}^n)}_{\text{Point. Out. (5)}}(t) = \sum_{i=1}^{P^n} g^n[i] p_t^{n,i} \quad (3)$$

where $g^n : \{1, \dots, P^n\} \rightarrow \mathbb{R}^{Z \times 1}$ is the pointwise filter, and the resulting convolution produces Z output channels, so $(g^n * \mathbf{p}^n)(t) \in \mathbb{R}^{Z \times 1}$.

3.4 Skip Connections

We propagate skip connections [20] to allow each layer to see the original data and the pointwise outputs from previous layers. This helps the network to cope with sparsely sampled data. For example, suppose a particular blood test is taken once per day. In order not to lose temporal resolution, we forward-fill these data (Section 4) and convolve with increasingly dilated temporal filters until we find the appropriate width to capture a useful trend. However, if the smaller filters in previous layers (which did not see any useful trend) have polluted the original data by re-weighting, learning

²We use a wider set of features for pointwise convolution, including static features \mathbf{s} and decay indicators \mathbf{x}'' i.e. $\mathbf{p}_t^n = (\mathbf{b}(\mathbf{h}_t^n) \parallel \mathbf{s} \parallel \mathbf{x}_t^n) \in \mathbb{R}^{P^n \times 1}$.

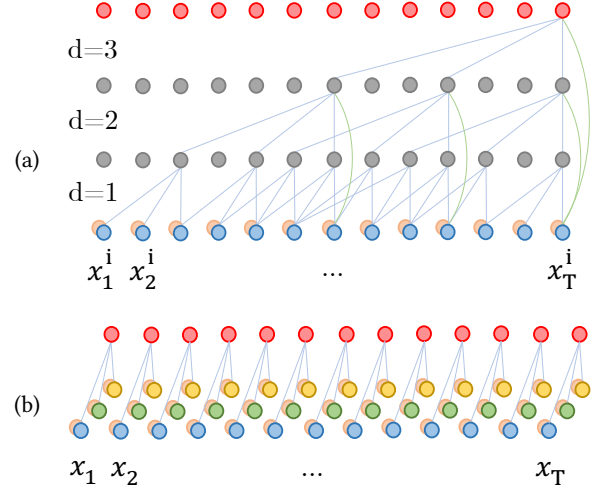


Figure 2: (a) Temporal convolution with skip connections (green lines). Each time series, i (blue dots) and their decay indicators (pale orange dots) are processed with independent parameters. (b) Pointwise convolution. There is no information sharing across time, only across features (blue, green, yellow dots).

will be harder. Therefore, skip connections provide a consistent anchor to the input. They are concatenated (like in DenseNet [22], and are arranged in the shared-source connection formation [64]) as illustrated in Figure 2. The skip connections expand the feature dimension, $R^n = F + Z(n-1)$, to accommodate the pointwise outputs, and also the channel dimension to fit the original data, $C^n = Y + 1$. This is best visualised in Figure 3.

3.5 Temporal Pointwise Convolution

Our model – which we refer to as Temporal Pointwise Convolution (TPC) – combines temporal and pointwise convolution in parallel. Firstly, the temporal output is combined with the skip connections to form \mathbf{r}_t^n (Step 3 in Figure 3).

$$\underbrace{\mathbf{r}_t^n}_{(3)} = \underbrace{(f^n * \mathbf{h}_t^n)}_{\text{Temp. Out. (2)}} \parallel \underbrace{\left[\left\| \left\| (g^{n'} * \mathbf{p}_t^{n'}) \right\| \right\|_{n'=1}^{n-1} \right]}_{\text{Skip Connections}} \quad (4)$$

\mathbf{r}_t^n is then concatenated with the pointwise output after it has been broadcast $Y + 1$ times. We can therefore define the n_{th} TPC layer as

$$\underbrace{\mathbf{h}_t^{(n+1)}}_{\text{TPC Out. (6)}} = \sigma \left(\underbrace{\mathbf{r}_t^n}_{(3)} \parallel \left[\left\| \left\| (g^n * \mathbf{p}_t^n) \right\| \right\|_{i=1}^{Y+1} \right] \right) \quad (5)$$

where σ represents the ReLU activation function. The full model has N TPC layers stacked sequentially. After N layers, the output \mathbf{h}_t^N is combined with static features $\mathbf{s} \in \mathbb{R}^{S \times 1}$, and a diagnosis embedding $\mathbf{d}^* \in \mathbb{R}^{D^* \times 1}$. Two pointwise layers are then applied to obtain the final predictions (see Appendix A for the full details). We use batch normalisation [23] and dropout [54] throughout to regularise the model.

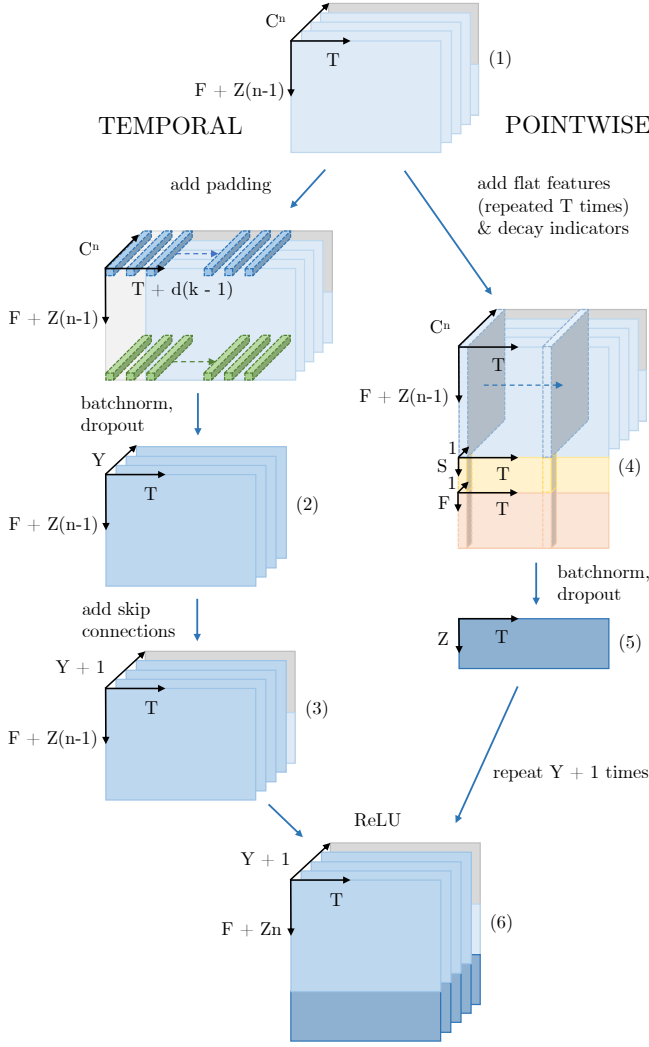


Figure 3: The n_{th} TPC layer. Left-sided padding (off-white) is added to the temporal side before each feature is processed independently. On the pointwise side, flat features (yellow) and decay indicators (orange) are added before each convolution.

3.6 Loss Function

The remaining LoS has a positive skew (shown in Figure 1) which makes the prediction task more challenging. We address this by replacing the commonly-used mean squared error (MSE) loss with mean squared log error (MSLE).

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T (\log(\hat{y}_t) - \log(y_t))^2 \quad (6)$$

MSLE penalises *proportional* errors, which is more reasonable when considering an error of e.g. 5 days in the context of a 2-day stay vs. a 30-day stay. The difference can be seen in Figure 4. For bed management purposes it is particularly important not to harshly penalise over-predictions – the model will become overly cautious and

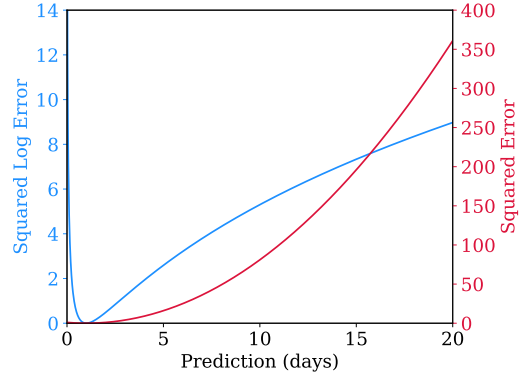


Figure 4: The behaviour of squared logarithmic error (blue) and squared error (red) functions when the true LoS is 1 day.

regress its predictions towards the mean. This is counter-productive because long stay patients have a disproportionate effect on bed occupancy.

4 DATA

4.1 eICU Database

We use the eICU Collaborative Research Database [39], a multi-centre dataset collated from 208 care centres in the United States, available through PhysioNet [14]. It comprises 200,859 patient unit encounters for 139,367 unique patients admitted to ICUs between 2014 and 2015.

We selected all adult patients (>18 years) with an ICU LoS of at least 5 hours and at least one recorded observation, resulting in 118,535 unique patients and 146,671 ICU stays. We selected 87 time series from the following tables: *lab*, *nursecharting*, *respiratorycharting*, *vitalperiodic* and *vitalaperiodic*. To be included, variables had to be present in at least 12.5% of patient stays, or 25% for *lab* variables. As shown in Figure 5, the *lab* variables tend to be sparsely sampled. To help the model cope with this missing data, we forward-filled over the gaps. This is more realistic than interpolation as the clinician would only have the most recent value. We then added ‘decay indicators’ to specify where the data is stale. **The decay was calculated as 0.75^j , where j is the time since the last recording. This is similar in spirit to the masking used by Che et al. [4].**

We extracted diagnoses from the *pasthistory*, *admissiondx* and *diagnoses* tables, and 17 static features from the *patient*, *apachepatientresult* and *hospital* tables (see Tables 5 and 16, and Appendix B for the full list of features and further details).

4.2 MIMIC-IV Database

We verify our results on a second dataset, the Medical Information Mart for Intensive Care (MIMIC-IV v0.4) database [24], a de-identified and publicly available EHR dataset from the Beth Israel Deaconess Medical Center containing 69,619 ICU stays from 50,048 patients admitted between 2008 and 2019.

We use the same cohort selection criteria as in eICU to select 69,609 ICU stays from 50,042 patients. We followed the same feature selection process to obtain a short list of 172 time series from the *chartevents* and *labevents*. We manually removed 71 of these from

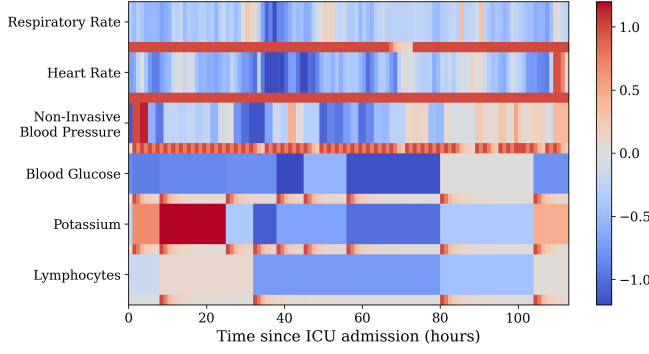


Figure 5: Example data from a patient in eICU (after pre-processing). The colour scale indicates the value of the feature, and the narrow bars show the corresponding decay indicators. Blood glucose, potassium and lymphocytes are from the *lab* table and are sparsely sampled. Non-invasive blood pressure is manually recorded by the nurse every 2 hours, while respiratory rate and heart rate are vital signs that are automatically logged.

chartevents because the variable did not vary over time, or because the distribution was not found to provide useful discrimination between patients (see Table 17 for the final list of features). We filled the missing data in the same way as in eICU. We extracted 12 flat features from the *icustays*, *admissions*, *patients* and *chartevents* tables (Table 6). We did not extract diagnoses from MIMIC-IV because they are not associated with reliable timestamps.

Table 1: Cohort summaries.

	eICU	MIMIC-IV
Number of patients	118,535	50,042
Train	82,973	35,028
Validation	17,781	7,507
Test	17,781	7,507
Number of stays	146,671	69,609
Train	102,749	48,848
Validation	22,033	10,497
Test	21,889	10,264
Gender (% male)	54.1%	55.8%
Age (mean)	63.1	64.7
LoS (mean)	3.01	3.98
LoS (median)	1.82	2.06
Remaining LoS (mean)	3.47	5.70
Remaining LoS (median)	1.67	2.70
In-hospital mortality	9.25%	11.4%
Number of input features	104	113
Time series	87	101
Static	17	12

5 EXPERIMENTS

In this section, we describe the prediction tasks, baseline models and evaluation metrics. As in Harutyunyan et al. [18] the training

and test data was fixed upfront – the patients were divided such that 70% were used for training, 15% for validation, and 15% for testing.

5.1 Prediction tasks

5.1.1 Remaining Length of Stay. We assign a remaining LoS target to each hour of the stay, beginning at 5 hours and ending when the patient dies or is discharged. We train the models to make a prediction every hour of the stay. We only include the first 14 days of any patient’s stay to protect against very long batches which would slow down training. This cut-off applies to <5% of patient stays, but it does *not* affect their maximum remaining LoS values.

5.1.2 In-Hospital Mortality. We also tested the performance of the models on mortality prediction. Unlike LoS, these labels remain static throughout the patient stay. We used the same training procedure as the LoS task i.e. one prediction each hour. However, to reflect the approach taken by Purushotham et al. [41] and Harutyunyan et al. [18], we only report the mortality performance once per patient (at 24 hours into the stay). This means that the cohort represented in the mortality metrics in Table 4 is smaller (16,239 of 21,889 test stays in eICU and 8,320 of 10,264 test stays in MIMIC-IV).

5.1.3 Multitask. Previous work has found merit in a multitask approach to patient outcome prediction [18, 51]. We investigated whether we would see a similar benefit in the TPC model. When combining the LoS and mortality losses, we applied a relative weighting to the mortality loss – dictated by a parameter α (which was treated as a hyperparameter). Further information on the hyperparameter search and implementation details is in Appendix C.

5.2 Baselines

We include the following baselines in our experiments:

‘Mean’ and ‘Median’ models (LoS only). These always predict 3.47 and 1.67 days respectively for eICU and 5.70 and 2.70 days for MIMIC-IV (these correspond to the mean and median of the training data). This is to benchmark the level of performance which is achievable ‘for free’ just by predicting in a reasonable range, and to provide points of reference when setting performance expectations for each dataset.

APACHE-IV values [67] (eICU only). These are generated by a risk assessment scoring model which is evaluated only once per patient at 24 hours. Therefore it cannot be compared directly, but we include it *only as a point of reference* for a widely used clinical model. APACHE-IV is only present in the eICU dataset.

Standard LSTM. Our standard LSTM is similar to Harutyunyan et al. [18].

Channel-wise LSTM (CW LSTM). Again similar to Harutyunyan et al. [18], this consists of a set of independent LSTMs that process each feature separately before concatenation (note the similarity with the independent temporal convolutions in the TPC model).

Transformer. This model takes advantage of multi-head self-attention. Like the TPC model, it is not constrained to progress one timestep at a time; however, unlike TPC, it is not able to scale its receptive fields or process features independently.

Table 2: Performance of the TPC model compared to baseline models. The loss function in all experiments is MSLE. For the first four metrics, lower is better. The error margins are 95% confidence intervals (CIs) calculated over 10 runs. These are not present for the mean, median and APACHE-IV models because they are deterministic. The best results are highlighted in blue. If the result is statistically significant on a t-test then it is indicated with stars (* $p < 0.05$, ** $p < 0.001$). MAD: mean absolute deviation; MAPE: mean absolute percentage error; MSE: mean squared error; MSLE: mean squared logarithmic error; R^2 : coefficient of determination, Kappa: Cohen Kappa Score. [†]Note that the APACHE-IV results (only present in the eICU dataset) cannot be compared directly to the other models (explained in Section 5.2).

Data	Model	MAD	MAPE	MSE	MSLE	R^2	Kappa
eICU	Mean	3.21	395.7	29.5	2.87	0.00	0.00
	Median	2.76	184.4	32.6	2.15	-0.11	0.00
	APACHE-IV [†]	2.54	182.1	16.6 [†]	1.10	-0.01	0.20
	LSTM	2.39±0.00	118.2±1.1	26.9±0.1	1.47±0.01	0.09±0.00	0.28±0.00
	CW LSTM	2.37±0.00	114.5±0.4	26.6±0.1	1.43±0.00	0.10±0.00	0.30±0.00
	Transformer	2.36±0.00	114.1±0.6	26.7±0.1	1.43±0.00	0.09±0.00	0.30±0.00
	TPC	1.78±0.02**	63.5±4.3**	21.7±0.5**	0.70±0.03**	0.27±0.02**	0.58±0.01**
MIMIC-IV	Mean	5.24	474.9	77.7	2.80	0.00	0.00
	Median	4.60	216.8	86.8	2.09	-0.12	0.00
	LSTM	3.68±0.02	107.2±3.1	65.7±0.7	1.26±0.01	0.15±0.01	0.43±0.01
	CW LSTM	3.68±0.02	107.0±1.8	66.4±0.6	1.23±0.01	0.15±0.01	0.43±0.00
	Transformer	3.62±0.02	113.8±1.8	63.4±0.5	1.21±0.01	0.18±0.01	0.45±0.00
	TPC	2.39±0.03**	47.6±1.4**	46.3±1.3**	0.39±0.02**	0.40±0.02**	0.78±0.01**

5.3 Evaluation Metrics

5.3.1 Length of Stay. We report on 6 LoS metrics: mean absolute deviation (MAD), mean absolute percentage error (MAPE), mean squared error (MSE), mean squared log error (MSLE), coefficient of determination (R^2) and Cohen Kappa Score. This is important because bad models can ‘cheat’ particular metrics just by being close to the mean or median value (see Appendix D for additional discussion on this).

5.3.2 In-Hospital Mortality. In the mortality and multitask experiments we report the area under the receiver operating characteristic curve (AUROC) and the area under the precision recall curve (AUPRC).

6 RESULTS

In this section, we analyse the model in several ways. Firstly, we report overall performance and compare against a set of baselines. Next, we examine the role of the loss function. Finally, we perform a set of ablation studies to find out which components of the model architecture contribute the most to its success.

6.1 TPC Performance on Length of Stay

The TPC model outperforms all of the baseline models on every metric on both datasets (Table 2) – particularly those that are more robust to skewness: MAPE, MSLE and Kappa. Discounting APACHE-IV, the best performing *baseline* across both datasets is the Transformer (although the *channel-wise* LSTM (CW LSTM) is similar on eICU). This is consistent with Harutyunyan et al. [18] (for CW LSTM) and Song et al. [53] (for Transformers), who found small improvements over standard LSTMs.

Performance differences between eICU and MIMIC-IV. Although the pattern of results is remarkably similar between eICU and MIMIC-IV, there are notable differences in the magnitudes of the metrics. These differences can be attributed to their LoS distributions – the positive skew is more severe in MIMIC-IV (Table 1). This skew has a disproportionate impact on the *absolute* error, which is captured in the MSE and MAD metrics. Interestingly, the Kappa score is higher in MIMIC-IV because the model can assign the longest stay patients to the >8 day bin, whereas eICU has more medium stay patients in the 3-8 day range which need to be precisely placed. The most comparable results are the MSLE and MAPE metrics, both of which penalise the *proportional* error, making them more robust to shifts in the LoS distribution.

6.2 Ablation Studies

To understand the impact of each design choice for the TPC model, we study performance under different ablations on the eICU dataset. The results of these ablations are reported in Table 3.

6.2.1 MSLE Loss Function. The first two rows of Table 3 show that using the MSLE (rather than MSE) loss function leads to significant improvements in the TPC model, with large performance gains in MAD, MAPE, MSLE and Kappa, while conceding little in terms of MSE and R^2 . The MSE results for the other models are in Appendix Table 13; they show a similar pattern to the TPC model.

6.2.2 Model Architecture. The second subtable shows that the temporal-only model is superior to the pointwise-only model, but neither reaches the performance of the TPC model. The temporal-only model performs much better than its weight-sharing variant, which demonstrates the importance of having independent parameters per feature. Note that the temporal-only model with weight sharing is the most similar to the approach taken by Razavian et al.

Table 3: Ablation studies of the TPC model (performed on the eICU dataset). Unless otherwise specified, the loss function is MSLE. The first subtable compares the effect of the loss function on the TPC model (see Table 13 in the Appendix for the MSE results of LSTM, CW LSTM and Transformer). The second shows various TPC ablation studies. Results that are not significantly different from the best result are highlighted in light blue. The TPC (MSLE) result has been repeated in each subtable for ease of comparison. WS: weight sharing; "no skip": no skip connections; "no diag.": no diagnoses, "no decay": no decay indicators.

Model	MAD	MAPE	MSE	MSLE	R^2	Kappa
TPC (MSLE)	1.78±0.02**	63.5±4.3**	21.7±0.5	0.70±0.03**	0.27±0.02	0.58±0.01**
TPC (MSE)	2.21±0.02	154.3±10.1	21.6±0.2	1.80±0.10	0.27±0.01	0.47±0.01
TPC	1.78±0.02	63.5±3.8*	21.8±0.5	0.71±0.03*	0.26±0.02	0.58±0.01
Point. only	2.68±0.15	137.8±16.4	29.8±2.9	1.60±0.03	-0.01±0.10	0.38±0.01
Temp. only	1.91±0.01	71.2±1.1	23.1±0.2	0.86±0.01	0.22±0.01	0.52±0.01
Temp. only (WS)	2.34±0.01	116.0±1.2	26.5±0.2	1.40±0.01	0.10±0.01	0.31±0.00
TPC (no skip)	1.93±0.01	73.9±1.9	23.0±0.2	0.89±0.01	0.22±0.01	0.51±0.01
TPC (no diag.)	1.77±0.02	65.6±4.1	21.5±0.5	0.71±0.03*	0.27±0.02	0.59±0.01
TPC (no decay)	1.84±0.01	64.5±3.0	22.5±0.3	0.77±0.02	0.24±0.01	0.56±0.01
Point. (no decay)	2.90±0.18	179.1±17.4	34.2±4.6	1.80±0.05	-0.16±0.16	0.33±0.00

Table 4: Performance of the TPC model in the multitask setting. We compare the performance of each model on individual tasks (mortality only on the first line; LoS only on the second) to the multitask setting (both LoS and mortality on the third line). The performance of the baseline models are reported in Tables 14 and 15.

Data	In-Hospital Mortality		MAD	MAPE	Length of Stay		R^2	Kappa
	AUROC	AUPRC			MSE	MSLE		
eICU	0.864±0.001	0.508±0.005	–	–	–	–	–	–
	–	–	1.78±0.02	63.5±3.8	21.8±0.5	0.71±0.03	0.26±0.02	0.58±0.01
	0.865±0.002	0.523±0.006**	1.55±0.01**	46.4±2.6**	18.7±0.2**	0.40±0.02**	0.37±0.01**	0.70±0.00**
MIMIC-IV	0.905±0.001	0.691±0.006	–	–	–	–	–	–
	–	–	2.39±0.03	47.6±1.4	46.3±1.3	0.39±0.02	0.40±0.02	0.78±0.01
	0.918±0.002**	0.713±0.007**	2.28±0.07*	32.4±1.2**	42.0±1.2**	0.19±0.00**	0.46±0.02**	0.85±0.00**

[45], and the results are comparable to the LSTM which is consistent with the results presented in the paper. Removing the skip connections reduces performance by 5-25%. Together the ablation studies demonstrate that the superior performance of the TPC model is the culmination of multiple design decisions.

6.2.3 Data. We also tested the models without the diagnoses or decay indicators. Perhaps surprisingly, we found that the exclusion of diagnoses does not seem to harm the model. This could be because the relevant diagnoses for predicting LoS e.g. Acute Respiratory Distress Syndrome (ARDS), are discernible from the time series alone e.g. PaO₂, FiO₂, PEEP etc. The decay indicators contribute a small (but statistically significant) benefit. Their contribution is more obvious in the pointwise-only model where all of the metrics see improvements of 5-23%. This difference is expected since they might reveal some of the temporal structure to the pointwise model e.g. reveal links between up-to-date observations and patient deterioration.

In Appendix E we tested the models without the laboratory tests (which are infrequently sampled) and without the other time series (which tend to be regularly monitored). They indicate that the TPC model is able to exploit disparate EHR time series more successfully than the baselines. They also show that the advantage of the CW

LSTM over the standard LSTM is only apparent when the model has to process different types of time series simultaneously.

6.3 Mortality and Multitask Performance

We investigated adding in-patient mortality as a side-task to improve LoS prediction. Table 4 shows the TPC performance both on *single-task* mortality prediction, as well as the multi-task setting. We observe first that TPC achieves good performance on mortality alone. Comparing the impact on LoS forecasting in the multi-task setting, we see significant improvements on every metric. Multi-task performance for all baselines is reported in Tables 14 and 15 in the Appendix, where the multitask training confers a more modest benefit.

7 FURTHER ANALYSES

In this section, we further explore the performance and behaviour of the TPC model for LoS prediction on the eICU dataset. We test its capacity to exploit smaller datasets, explore which features it uses, and provide a visualisation of the reliability of the model. Finally, we simulate the potential use of the model for bed planning.

7.1 Training Data Size

The TPC model consistently outperforms the baselines when the training data is small, but we noticed even greater potential for big data. We tested the TPC, LSTM, CW LSTM, and Transformer models with 6.25%, 12.5%, 25%, 50%, and 100% of the eICU training data. TPC maintains the best test performance on all data sizes, with an increasing benefit for larger data. Figure 6 shows the effect on MSLE (the full results for all metrics are included in Table 12).

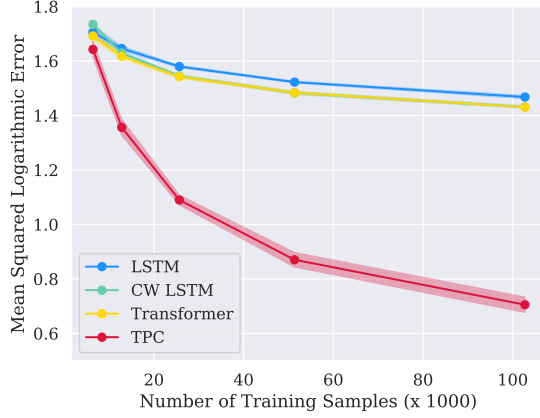


Figure 6: The effect of changing the training data size on the LSTM, CW LSTM, Transformer, and TPC model performance on the eICU dataset. Only the mean squared logarithmic error (MSLE) is shown for clarity, however the other metrics are shown in Table 12. Note that the performance of the CW LSTM and Transformer models are so similar that the curves are superimposed.

7.2 Feature Importance

We used the integrated gradients method [56] to calculate feature attributions for the LoS estimates in the eICU dataset. This method computes the importance scores ϕ_i^{IG} by accumulating gradients interpolated between a baseline input \mathbf{b} (intended to represent the absence of data) and the current input \mathbf{x} :

$$\phi_i^{IG}(\psi, \mathbf{x}, \mathbf{b}) = \underbrace{(\mathbf{x}_i - \mathbf{b}_i)}_{\text{diff. from baseline}} \times \int_{\alpha=0}^1 \underbrace{\frac{\delta\psi(\mathbf{b} + \alpha(\mathbf{x} - \mathbf{b}))}{\delta\mathbf{x}_i}}_{\text{acc. local grad.}} d\alpha \quad (7)$$

where the TPC model is represented as ψ . We use the mean feature values as our baseline input vector. We take the absolute attribution values when a single LoS prediction is made for each patient at 24 hours. We aggregate by taking the mean along the time dimension and then the patient dimension to obtain Figure 7. The background and intuition behind the method is explained clearly by Sturmfels et al. [55].

Analysing Figure 7, we note that the top features are all strong indicators of organ failure: troponin I is a specific biomarker of myocardial infarction; peak inspiratory pressure, O₂ L/%, TV/kg IBW, plateau pressure, PEEP and tidal volume indicate mechanical ventilation (on account of respiratory failure); PTT, ALT (SGPT), AST (SGOT) and alkaline phosphatase suggest liver disease; and

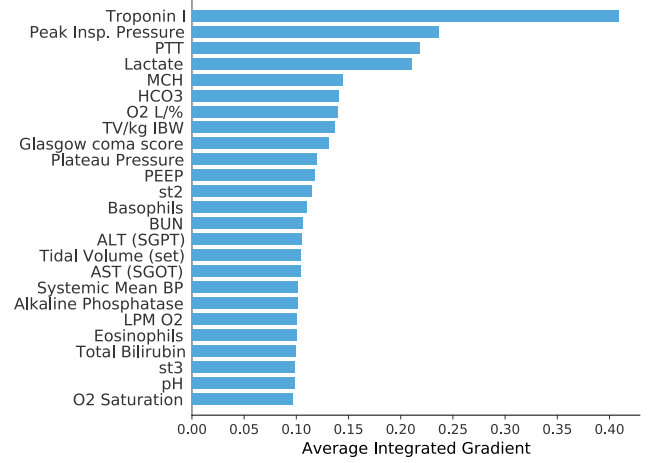


Figure 7: Top 25 most important features to the TPC model in the eICU dataset.

high BUN and bilirubin levels point towards kidney failure. Additionally we see infection markers such as lactate, basophils and eosinophils which could indicate sepsis. Both multi-organ failure and sepsis are known causes of extended LoS in the ICU [2].

7.3 Evaluation by Use-Case

We have reported aggregate performance metrics indicating strong performance of the TPC model for overall LoS forecasting. In this section, we provide further evaluations tailored to two potential users – an individual ICU clinician, and a bed manager for the unit.

7.3.1 Individual-level Reliability. Although aggregate measures of performance are typically reported, these can mask underlying variability in model performance. Such variability can undermine trust or result in unsafe application of systems [50]. In this section, we think of a clinician who wishes to interpret the prediction of the system for an *individual* patient. We break down the aggregate performance metrics based on factors which will be readily-available at the time of the prediction. Specifically, we visualise the MAPE (chosen for its interpretability) as a function of the time since admission and the *predicted* remaining LoS.

Figure 8 shows an example for the TPC model on eICU. We can see that high predicted remaining LoS on the *first* day of a patient's stay can be quite unreliable, with performance rapidly improving over time. Additional investigation revealed these initial predictions to be *under*-predictions, indicating that it is challenging to accurately forecast *very long* LoS for patients on their first day. The long tail of LoS in the dataset reflects the abundance of short-stay patients. The model therefore seems to wait for 1-2 days of data to justify a long LoS prediction. The system can therefore be equipped with instructions indicating that a high predicted remaining LoS on the *first* day should not be acted upon. This could complement information provided on a model card [33, 50].

7.3.2 ICU-level Bed Management. From the perspective of a bed manager, *aggregate* performance of the model is important: an over-prediction for one patient could be offset by an under-prediction for

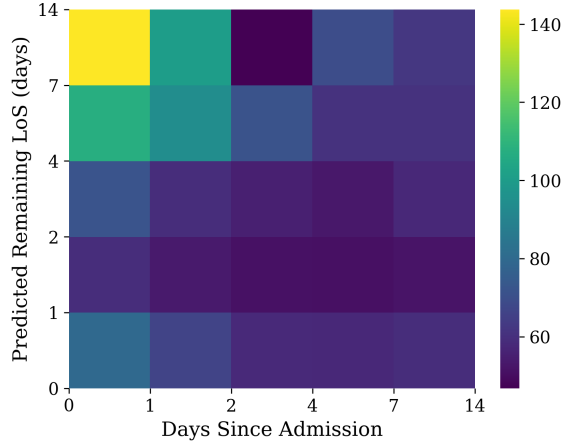


Figure 8: Mean absolute percentage error as a function of days since admission and predicted remaining LoS on the eICU dataset.

another, resulting in the same net bed availability. To investigate this, we performed a simulation study. We ran 500 ICU simulations by randomly selecting 16 examples from the eICU test set to form a ‘virtual cohort’. The number 16 was chosen because US hospitals have, on average, 24 ICU beds [63] with an occupancy rate of 68% [17]. Figure 9 shows the number of patients remaining in the ICU (of the selected cohort; we do not visualise incoming ICU admissions) using their true remaining LoS (blue). We compute the error (red) between the predictions (green) and true values. The model is well calibrated when predicting patients who are going to stay for at least 1 day. After this, the model tends to under-predict the occupancy by approximately 0.8 patients, corresponding to a small bias towards under-estimating the remaining LoS.

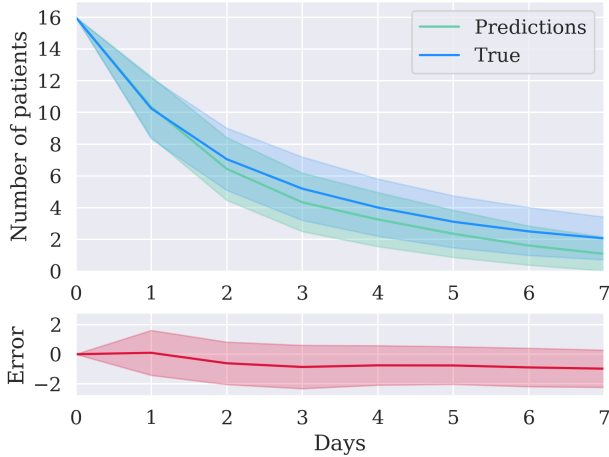


Figure 9: ICU simulation. We show the number of patients remaining in the ICU over time from an initial cohort of 16 random eICU patients from 500 simulations. The shaded regions show the standard deviation across the runs. ‘Error’ is calculated from ‘True’ minus ‘Predictions’.

8 DISCUSSION

We have shown that the TPC model outperforms all baseline models in all task settings (LoS, mortality or multitask) on both the eICU and MIMIC-IV datasets. To explain the success of TPC, we start by examining the parallel architectures in the TPC model. Each component has been designed to extract different information: trends from the temporal convolutions and inter-feature relationships from the pointwise convolutions. The eICU ablation studies reveal that the temporal element is more important, but we stress that their contributions are complementary since the best performance is achieved when they are used together.

Next, we highlight that the temporal-only model far outperforms its most direct comparison, the CW LSTM, on all metrics. Theoretically, they are well matched because they both have feature-specific parameters but are restricted from learning cross-feature interactions. To begin to explain this, we consider how the information flows through the model. The temporal-only model can directly step across large time gaps, whereas the CW LSTM is forced to progress one timestep at a time. This gives the CW LSTM the harder task of remembering information across a noisy EHR with distracting signals of varying frequency. In addition, the temporal-only model can tune its receptive fields for improved processing of each feature thanks to the skip connections (which are not present in the CW LSTM).

The difference in performance between the temporal-only model with and without weight sharing provides strong evidence that assigning independent parameters to each feature is important. Some EHR time series are irregularly and sparsely sampled, and can exhibit considerable variability in the temporal frequencies within the underlying data (evident in Figure 5). This presents a challenge for any model, especially if it is constrained to learn one set of parameters to suit all features. The relative success of the CW LSTM over the standard LSTM when processing *disparate* time series – but not similar – also lends weight to this theory.

However, the assignment of independent parameters to each feature does not explain all the successes of TPC e.g. the TPC model can process disparate time series and gain more marginal performance than the CW LSTM (Table 11). We need to consider that *periodicity* is a key property of EHR data – this is true in both the sampling patterns and in the underlying biology e.g. medication schedules, sleep cycles, meals etc. The temporal component of the TPC model is the only architecture with an inherent periodic structure (from the stacked temporal filters) which makes it much easier to learn EHR trends. By comparison, a single attention head in the Transformer model does not look at timepoints a fixed distance apart, but can take an arbitrary form. This is a strength for natural language processing, given the variety of sentence structures possible, but it does not help the Transformer to process EHRs.

Additionally, we have shown that the TPC model outperforms baselines on in-hospital mortality both as a standalone task and in combination with LoS. The performance on both mortality and LoS is significantly better in the multitask setting (this is consistent with past works [18, 51]) because multitask learning helps to regularise the model and reduce the chance of overfitting [48]. Adding further tasks may be a valid strategy to improve LoS performance.

Finally, we reiterate that using MSLE loss instead of MSE greatly mitigates for positive skew in the LoS task, and this benefit is not model-specific (all of the baselines perform better with MSLE – see Table 13). This demonstrates that careful consideration of the task – as well as the data and model – is an important step towards building useful tools in healthcare.

8.1 Limitations and Future Work

Our work has several limitations. We know that LoS is heavily influenced by operational factors, and clinical practices can change over time [26]. Capacity to maintain performance over time is an important consideration before a system could be used in practice. In future work, it would be instructive to test how quickly the models become out-of-date by reserving more recent data as a test set [35]. Although we have included a large set of baselines, we acknowledge that a more exhaustive comparison could be performed, for example comparing by Gaussian Processes [40] or ODE-RNNs [9, 47] for handling irregularly sampled time-series. Finally, although we have motivated our study by bed management, this work describes a methodological proof of concept and does not constitute a real clinical system. Prospective study and integration into a real-world EHR is necessary to demonstrate real-world benefit, both of which pose their own challenges [43, 49].

In future work, we would like to investigate why the TPC model gains more from the multitask setting than the other models. It seems likely that it is related to additional regularisation provided by the mortality task, but further investigation is needed to confirm our speculations.

9 CONCLUSION

We have proposed and evaluated a new deep learning architecture, which we call ‘Temporal Pointwise Convolution’ (TPC). TPC combines temporal convolutional layers with pointwise convolutions to extract temporal and inter-feature information. We have shown that the TPC model is well-equipped to analyse EHR time series containing missingness, differing frequencies and sparse sampling. We believe that the following four aspects contribute the most to its success:

- (1) The combination of two complementary architectures that are able to extract different features, both of which are important.
- (2) The ability to step over large time gaps.
- (3) The capacity to specialise processing to each feature (including the freedom to select the receptive field size for each).
- (4) The rigid spacing of the temporal filters, making it easy to derive trends.

From a clinical perspective, we have contributed to the advancement of LoS prediction models, a prerequisite for automated bed management tools. Improving the practice of bed management promises cost reduction [17] and better resource allocation [32] worldwide. From a computational perspective, we have provided key insights for retrospective EHR studies, particularly where LSTMs are the currently model of choice. In the broader context of machine learning for healthcare we have demonstrated that careful consideration of the complexities of health data is necessary to gain state-of-the-art performance in these tasks.

ACKNOWLEDGEMENTS

The authors would like to thank Alex Campbell, Petar Veličković, and Ari Ercole for helpful discussions and advice. We would also like to thank Louis-Pascal Xhonneux, Seyon Sivarajah, Rudolf Cardinal, Jacob Deasy, Paul Scherer, and Katharina Kohler for their help in reviewing the manuscript. Finally we thank the Armstrong Fund, the Frank Edward Elmore Fund, and the School of Clinical Medicine at the University of Cambridge for their generous funding.

REFERENCES

- [1] Mathias C Blom, Karin Erwarder, Lars Gustafsson, Mona Landin-Olsson, Fredrik Jonsson, and Kjell Ivarsson. 2015. The probability of readmission within 30 days of hospital discharge is positively associated with inpatient bed occupancy at discharge – a retrospective cohort study. *BMC Emergency Medicine* 15, 1 (2015), 37. <https://doi.org/10.1186/s12873-015-0067-9>
- [2] Andreas B Böhmer, Katja S Just, Rolf Lefering, Thomas Paffrath, Bertil Bouillon, Robin Joppich, Frank Wappler, and Mark U Gerbershagen. 2014. Factors influencing lengths of stay in the intensive care unit for surviving trauma patients: a retrospective analysis of 30,157 cases. *Critical care (London, England)* 18, 4 (2014), R143–R143.
- [3] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. BRITS: Bidirectional Recurrent Imputation for Time Series. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc.
- [4] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports* 8, 1 (2018), 6085.
- [5] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2015. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. *JMLR workshop and conference proceedings* 56 (2015), 301–318.
- [6] F. Chollet. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 1800–1807.
- [7] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46.
- [8] Deborah Dahl, Greg G Wojtal, Michael Breslow, Randy Holl, Debra Huguez, David Stone, and Gloria Korpi. 2012. The High Cost of Low-Acuity ICU Outliers. *Journal of healthcare management / American College of Healthcare Executives* 57 (2012), 421–434.
- [9] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. 2019. GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.
- [10] Thuppathi Sisira De Silva, Don MacDonald, Grace Paterson, Khokan C. Sikdar, and Bonnie Cochrane. 2011. Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) to Represent Computed Tomography Procedures. *Comput. Methods Prog. Biomed.* 101, 3 (2011), 324–329.
- [11] A Elixhauser, C Steiner, and L Palmer. 2015. Clinical Classifications Software.
- [12] Kunihiko Fukushima. 1980. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics* 36, 4 (1980), 193–202.
- [13] T. Gentimis, A. J. Alnaser, A. Durante, K. Cook, and R. Steele. 2017. Predicting Hospital Length of Stay Using Neural Networks on MIMIC III Data. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing*. 1194–1201.
- [14] A. L. Goldberger, L. A. N. Amaral, L. Glass, et al. 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101, 23 (2000), e215–e220.
- [15] Jen J. Gong, Tristan Naumann, Peter Szolovits, and John V. Guttag. 2017. Predicting Clinical Outcomes Across Changing Electronic Health Record Systems. In *KDD*.
- [16] Çağlar Gülçehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. 2016. Noisy Activation Functions. *CoRR abs/1603.00391* (2016). arXiv:1603.00391
- [17] Neil A Halpern and Stephen M Pastores. 2015. Critical Care Medicine Beds, Use, Occupancy, and Costs in the United States: A Methodological Review. *Critical care medicine* 43, 11 (2015), 2452–2459.
- [18] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask Learning and Benchmarking with Clinical Time Series Data. *Scientific Data* 6, 96 (2019).
- [19] Mahmud Hassan, Howard Tuckman, Robert Patrick, David Kountz, and Jennifer Kohn. 2010. Hospital Length of Stay and Probability of Acquiring Infection. *International Journal of Pharmaceutical and Healthcare Marketing* 4 (2010), 324–338.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR abs/1512.03385* (2015). arXiv:1512.03385
- [21] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation* 9, 8 (1997), 1735–1780.

- [22] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- [23] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* (Lille, France) (ICML '15). JMLR, 448–456.
- [24] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark. 2020. Medical Information Mart for Intensive Care IV. <https://doi.org/10.13026/a3wn-hq05>.
- [25] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural Machine Translation in Linear Time. *CoRR* abs/1610.10099 (2016). [arXiv:1610.10099](https://arxiv.org/abs/1610.10099)
- [26] Amit D. Kalra, Robert S Fisher, and Peter Axelrod. 2010. Decreased Length of Stay and Cumulative Hospitalized Days despite Increased Patient Admissions and Readmissions in an Area of Urban Poverty. *J Gen Intern Med* 25, 9 (2010), 920–935.
- [27] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).
- [28] Kevin B. Laupland, Andrew W. Kirkpatrick, John B. Kortbeek, and Danny J. Zuege. 2006. Long-term Mortality Outcome Associated With Prolonged Admission to the ICU. *Chest* 129, 4 (2006), 954–959.
- [29] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network In Network. [arXiv:1312.4400 \[cs.NE\]](https://arxiv.org/abs/1312.4400)
- [30] Zachary Chase Lipton, David C. Kale, Charles Elkan, and Randall C. Wetzel. 2015. Learning to Diagnose with LSTM Recurrent Neural Networks. *CoRR* abs/1511.03677 (2015).
- [31] Gregory Mak, William D. Grant, James C McKenzie, and John B. McCabe. 2012. Physicians' Ability to Predict Hospital Length of Stay for Patients Admitted to the Hospital from the Emergency Department. In *Emergency medicine international*.
- [32] Kusum S Mathews and Elisa F Long. 2015. A Conceptual Framework for Improving Critical Care Patient Flow and Bed Use. *Annals of the American Thoracic Society* 12, 6 (2015), 886–894.
- [33] Margaret Mitchell, Simone Wu, A. Zaldivar, P. Barnes, Lucy Vasserman, B. Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency* (2019).
- [34] S Mostafa Mousavi, William L Ellsworth, Weiqiang Zhu, Lindsay Y Chuang, and Gregory C Beroza. 2020. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications* 11, 1 (2020), 3952. <https://doi.org/10.1038/s41467-020-17591-w>
- [35] Bret Nestor, Matthew B. A. McDermott, Geeticka Chauhan, Tristan Naumann, Michael C. Hughes, Anna Goldenberg, and Marzyeh Ghassemi. 2018. Rethinking Clinical Prediction: Why Machine Learning must Consider Year of Care and Feature Aggregation. *CoRR* abs/1811.12583 (2018). [arXiv:1811.12583](https://arxiv.org/abs/1811.12583)
- [36] NHS Digital. 2019. DCB0084: OPCS-4.9 Requirements Specification.
- [37] Jeeheh Oh, Jiaxuan Wang, and Jenna Wiens. 2018. Learning to Exploit Invariances in Clinical Time-Series Data using Sequence Transformer Networks. In *Proceedings of the 3rd Machine Learning for Healthcare Conference (Proceedings of Machine Learning Research, Vol. 85)*, Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens (Eds.). PMLR, Palo Alto, California, 332–347. <http://proceedings.mlr.press/v85/oh18a.html>
- [38] Adam Paszke, Sam Gross, Francisco Massa, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035.
- [39] Tom J Pollard, Alistair E W Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. 2018. The eICU Collaborative Research Database, A Freely Available Multi-Center Database for Critical Care Research. *Scientific Data* 5, 1 (2018), 180178.
- [40] Niranjani Prasad, Li-Fang Cheng, Corey Chivers, Michael Draugelis, and Barbara E. Engelhardt. 2017. A Reinforcement Learning Approach to Weaning of Mechanical Ventilation in Intensive Care Units. *CoRR* abs/1704.06300 (2017). [arXiv:1704.06300](https://arxiv.org/abs/1704.06300)
- [41] Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. 2017. Benchmark of Deep Learning Models on Large Healthcare MIMIC Datasets. [arXiv:1710.08531 \[cs.LG\]](https://arxiv.org/abs/1710.08531)
- [42] S. Purushotham, C. Meng, Z. Che, and Y. Liu. 2018. Benchmarking Deep Learning Models on Large Healthcare Datasets. *Journal of Biomedical Informatics* 83 (2018), 112–134.
- [43] Alvin Rajkomar, Eyal Oren, Kai Chen, et al. 2018. Scalable and Accurate Deep Learning with Electronic Health Records. In *npj Digital Medicine*.
- [44] John Rapoport, Daniel Teres, Yonggang Zhao, and Stanley Lemeshow. 2003. Length of Stay Data as a Guide to Hospital Economic Performance for ICU Patients. *Medical Care* 41 (2003), 386–397.
- [45] Narges Razavian, Jake Marcus, and David Sontag. 2016. Multi-task Prediction of Disease Onsets from Longitudinal Laboratory Tests (*Proceedings of Machine Learning Research, Vol. 56*), Finale Doshi-Velez, Jim Fackler, David Kale, Byron Wallace, and Jenna Wiens (Eds.). PMLR, Northeastern University, Boston, MA, USA, 73–100. <http://proceedings.mlr.press/v56/Razavian16.html>
- [46] Narges Razavian and David A. Sontag. 2015. Temporal Convolutional Neural Networks for Diagnosis from Lab Tests. *CoRR* abs/1511.07938 (2015). [arXiv:1511.07938](https://arxiv.org/abs/1511.07938)
- [47] Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. 2019. Latent ODEs for Irregularly-Sampled Time Series. *NeurIPS* abs/1907.03907 (2019).
- [48] Sebastian Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR* abs/1706.05098 (2017). [arXiv:1706.05098](https://arxiv.org/abs/1706.05098)
- [49] Mark Sendak, Madeleine Clare Elish, Michael Gao, Joseph Futoma, William Ratliff, Marshall Nichols, Armando Bedoya, Suresh Balu, and Cara O'Brien. 2020. "The Human Body is a Black Box": Supporting Clinical Decision-Making with Deep Learning. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) (FAT* '20). Association for Computing Machinery, New York, NY, USA, 99–109. <https://doi.org/10.1145/3351095.3372827>
- [50] Mark P Sendak, Michael Gao, Nathan Brajer, and Suresh Balu. 2020. Presenting machine learning model information to clinical end users with model facts labels. *npj Digital Medicine* 3, 1 (2020), 41. <https://doi.org/10.1038/s41467-020-0253-3>
- [51] Seyedmostafa Sheikhalishahi, Vevake Balaraman, and Venet Osmani. 2019. Benchmarking Machine Learning Models on eICU Critical Care Dataset. [arXiv:1910.00964 \[cs.LG\]](https://arxiv.org/abs/1910.00964)
- [52] Benjamin Shickel, Tyler J. Loftus, Lasith Adhikari, Tezcan Ozzragat-Baslanti, Azra Bihorac, and Parisa Rashidi. 2019. DeepSOFA: A Continuous Acuity Score for Critically Ill Patients using Clinically Interpretable Deep Learning. In *Scientific Reports*.
- [53] Huan Song, Deepta Rajan, Jayaraman J. Thiagarajan, and Andreas Spanias. 2018. Attend and Diagnose: Clinical Time Series Analysis using Attention Models. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. 4091–4098.
- [54] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958.
- [55] Pascal Sturmels, Scott Lundberg, and Su-In Lee. 2020. Visualizing the Impact of Feature Attribution Baselines. *Distill* 5, 1 (2020), e22. <https://distill.pub/2020/attribution-baselines/>
- [56] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) (ICML '17). JMLR.org, 3319–3328.
- [57] Harini Suresh, Nathan Hunt, Alistair E. W. Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. 2017. Clinical Intervention Prediction and Understanding with Deep Neural Networks. In *MLHC*.
- [58] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going Deeper with Convolutions. *CoRR* abs/1409.4842 (2014). [arXiv:1409.4842](https://arxiv.org/abs/1409.4842)
- [59] Nenad Tomašev, Xavier Glorot, Jack W Rae, et al. 2019. A Clinically Applicable Approach to Continuous Prediction of Future Acute Kidney Injury. *Nature* 572, 7767 (2019), 116–119.
- [60] Sana Tonekaboni, Mjaye Mazwi, Peter Laussen, Danny Eytan, Robert Greer, Sebastian D. Goodfellow, Andrew Goodwin, Michael Brudno, and Anna Goldenberg. 2018. Prediction of Cardiac Arrest from Physiological Signals in the Pediatric ICU. In *MLHC*.
- [61] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. *CoRR* abs/1609.03499 (2016). [arXiv:1609.03499](https://arxiv.org/abs/1609.03499)
- [62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS '17)*. Curran Associates Inc., 6000–6010.
- [63] David J. Wallace, Derek C. Angus, Christopher W. Seymour, Amber E. Barnato, and Jeremy M. Kahn. 2015. Critical Care Bed Growth in the United States. A Comparison of Regional and National Trends. *American Journal of Respiratory and Critical Care Medicine* 191, 4 (2015), 410–416. PMID: 25522054.
- [64] Zhongyuan Wang, Peng Yi, Kui Jiang, Junjun Jiang, Zhen Han, Tao Lu, and Jiayi Ma. 2018. Multi-Memory Convolutional Neural Network for Video Super-Resolution. *IEEE Transactions on Image Processing* PP (12 2018), 1–1. <https://doi.org/10.1109/TIP.2018.2887017>
- [65] World Health Organisation. 2011. *ICD-10: International Statistical Classification of Diseases and Related Health Problems*. Vol. 10th Revision. World Health Organisation.
- [66] David Zimmerer, Jens Petersen, Gregor Köhler, Jakob Wasserthal, Tim Adler, Sebastian Winkert, and Tobias Ross. 2017. trixi - Training and Retrospective Insight eXperiment Infrastructure. <https://github.com/MIC-DKFZ/trixi>. <https://doi.org/10.5281/zenodo.1345136>
- [67] Jack E Zimmerman, Andrew A Kramer, Douglas S McNair, and Fern M Malila. 2006. Acute Physiology and Chronic Health Evaluation (APACHE) IV: Hospital Mortality Assessment for Today's Critically Ill Patients. *Read Online: Critical Care Medicine | Society of Critical Care Medicine* 34, 5 (2006).

A MODEL ARCHITECTURE: FURTHER DETAILS

After N TPC layers, we apply two further pointwise convolutions to obtain the final predictions. Formally, these final steps (shown in Figure 10) can be written as

$$\hat{y}_t = \text{HardTanh} \left(\exp \left(\underbrace{g'' * \sigma \left(g' * \left[b(\mathbf{h}_t^N) \parallel \mathbf{s} \parallel \mathbf{d}^* \right] \right)}_{\text{Penultimate Point. Out. (8)}} \right) \right) \quad (8)$$

(9)

where $B = R^N \times (Y + 1) + S + D^*$ and the final pointwise filters are $g' : \{1, \dots, B\} \rightarrow \mathbb{R}^{X \times 1}$ and $g'' : \{1, \dots, X\} \rightarrow \mathbb{R}^{1 \times 1}$. Note that if a baseline model were to be used instead of TPC, the output dimensions would be $H \times 1$ instead of $B \times 1$, where H is the LSTM hidden size or d_{model} in the Transformer. We apply an exponential function to allow the upstream model to predict $\log(\text{LoS})$ instead of LoS. We hypothesised that this could help to circumvent a common issue seen in previous models (e.g. Harutyunyan et al. [18], as they struggle to produce predictions over the full dynamic range of length of stays). Finally, we apply a HardTanh function [16] to clip any predictions that are smaller than 30 minutes or larger than 100 days, which protects against inflated MSLE loss values.

$$\text{HardTanh}(x) = \begin{cases} 100, & \text{if } x > 100, \\ \frac{1}{48}, & \text{if } x < \frac{1}{48}, \\ x, & \text{otherwise.} \end{cases} \quad (9)$$

B FEATURE PRE-PROCESSING

B.1 Static Features

We selected 17 static features from eICU (shown in Table 5) and 12 from MIMIC-IV (Table 6). Discrete and continuous variables were scaled to the interval $[-1, 1]$, using the 5th and 95th percentiles as the boundaries, and absolute cut offs were placed at $[-4, 4]$. This was to protect against large or erroneous inputs, while avoiding assumptions about the variable distributions. Binary variables were coded as 1 and 0. Categorical variables were converted to one-hot encodings.

B.2 Diagnoses

Here we only describe pre-processing for eICU since MIMIC-IV did not contain coded diagnoses with appropriate timestamps.

Like many EHRs, diagnosis coding in eICU is hierarchical. At the lowest level they can be quite specific e.g. “neurologic | disorders of vasculature | stroke | hemorrhagic stroke | subarachnoid hemorrhage | with vasospasm”. To maintain the hierarchical structure within a flat vector, we assigned separate features to each hierarchical level and use binary encoding. This produces a vector of size 4,436 with an average sparsity of 99.5% (only 0.5% of the data is positive). We apply a 1% prevalence cut-off on all these features to reduce the size of the vector to 293 and the average sparsity to 93.3%. If a disease does not make the cut-off for inclusion, it is still included via any parent classes that do make the cut-off (in the

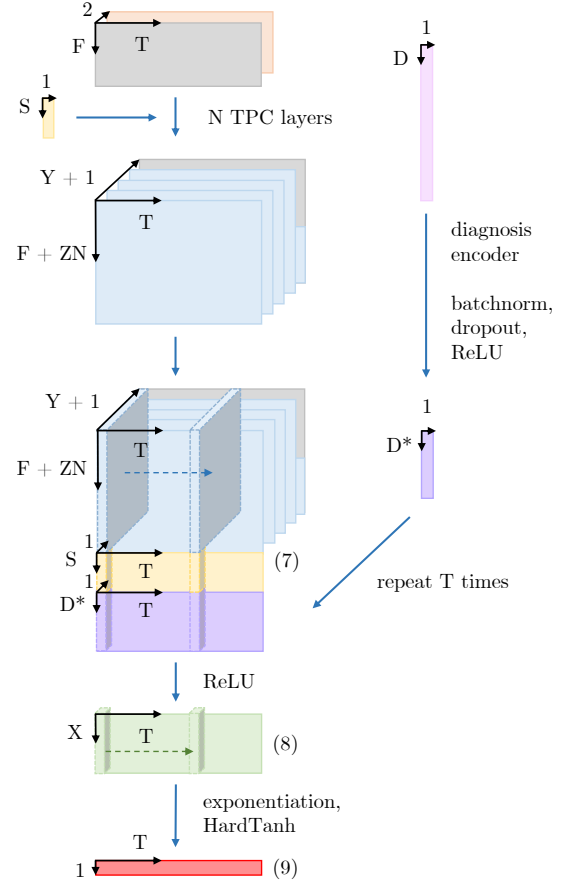


Figure 10: The original time series, x' (grey) and the decay indicators, x'' (orange) are processed by N TPC layers before being combined with a diagnosis embedding d^* (purple) and static features s (yellow) along the feature axis. A two-layer pointwise convolution is applied to obtain the final predictions \hat{y} (red).

above example we record everything up to “subarachnoid hemorrhage”). We only included diagnoses that were recorded before the 5th hour in the ICU, to avoid leakage from the future.

Many diagnostic and interventional coding systems are hierarchical in nature: ICD-10 classification [65], Clinical Classifications Software [11], SNOMED CT [10] and OPCS Classification of Interventions and Procedures [36], so this technique is generalisable to other coding systems present in EHRs.

B.3 Time Series

For each admission, we extracted 87 time-varying features from eICU (Table 16) and 101 from MIMIC-IV (Table 17) for each hour of the ICU visit, and up to 24 hours before the ICU visit. The variables were processed in the same manner as the static features. In general, the sampling is very irregular, so the data was re-sampled according to one hour intervals and forward-filled. After forward-filling is complete, any data recorded before the ICU admission is removed. Decay indicators are added as described in Section 4.

Table 5: eICU static features. Age >89, Null Height and Null Weight were added as indicator variables to indicate when the age was more than 89 but has been capped, and when the height or weight were missing and have been imputed with the mean value.

Feature	Type	Source Table
Gender	Binary	<i>patient</i>
Age	Discrete	<i>patient</i>
Hour of Admission	Discrete	<i>patient</i>
Height	Continuous	<i>patient</i>
Weight	Continuous	<i>patient</i>
Ethnicity	Categorical	<i>patient</i>
Unit Type	Categorical	<i>patient</i>
Unit Admit Source	Categorical	<i>patient</i>
Unit Visit Number	Categorical	<i>patient</i>
Unit Stay Type	Categorical	<i>patient</i>
Num Beds Category	Categorical	<i>hospital</i>
Region	Categorical	<i>hospital</i>
Teaching Status	Binary	<i>hospital</i>
Physician Speciality	Categorical	<i>apachepatientresult</i>
Age >89	Binary	
Null Height	Binary	
Null Weight	Binary	

Table 6: MIMIC-IV static features. Age was calculated from the ‘intime’ field in the *icustays* table and ‘anchor year’ in the *patients* table.

Feature	Type	Source Table
Gender	Binary	<i>patients</i>
Ethnicity	Categorical	<i>admissions</i>
Admission Location	Categorical	<i>admissions</i>
Insurance Type	Categorical	<i>admissions</i>
First Careunit	Categorical	<i>icustays</i>
Hour of Admission	Discrete	<i>icustays</i>
Admission Height	Continuous	<i>chartevents</i>
Admission Weight	Continuous	<i>chartevents</i>
Eyes	Discrete	<i>chartevents</i>
Motor	Discrete	<i>chartevents</i>
Verbal	Discrete	<i>chartevents</i>
Age	Discrete	

C HYPERPARAMETER SEARCH METHODOLOGY AND IMPLEMENTATION DETAILS

The TPC model and baselines have hyperparameters that can broadly be split into three categories: time series specific, non-time series specific and global parameters (shown in more detail in Tables 7, 8 and 9). The hyperparameter search ranges have been included in Table 10.

First, we ran 25 randomly sampled hyperparameter trials on the TPC model to decide the non-time series specific parameters

Table 7: The TPC model has 11 hyperparameters (Main Dropout and Batch Normalisation have been repeated in the table because they apply to multiple parts of the model). We allowed the model to optimise a custom dropout rate for the temporal convolutions because they have fewer parameters and might need less regularisation than the rest of the model. The best hyperparameter values are shown in brackets (eICU/MIMIC-IV). Hyperparameters marked with * were fixed across all of the models.

TPC Specific	
Temporal Specific	Pointwise Specific
Temp. Channels (12/11)	Point. Channels (13/5)
Temp. Dropout (0.05/0.05)	Main Dropout* (0.45/0)
Kernel Size (4/5)	
Batch Normalisation* (True/True)	
No. TPC Layers (9/8)	
Non-TPC Specific	Global Parameters
Diag. Embedding Size* (64/-)	Batch Size (32/8)
Main Dropout* (0.45/0)	Learning Rate (0.00226/0.00221)
Final FC Layer Size* (17/36)	
Batch Normalisation* (True/True)	

(diagnosis embedding size, final fully connected layer size, batch normalisation strategy, dropout rate and the parameter α) keeping all other parameters fixed. These parameters (indicated by stars) remained fixed for all the models which share their non-time series specific architecture (NB. the best value for α was 100 – not shown in the Tables).

We then ran 50 hyperparameter trials to optimise the remaining parameters for the TPC, standard LSTM, and Transformer models. To train the channel-wise LSTM and the temporal model with weight sharing, we ran a further 10 trials to re-optimize the hidden size (8 per feature) and number of temporal channels (32 channels shared across all features) respectively. For all other ablation studies and variations of each model, we kept the same hyperparameters where applicable (see Table 2 for a full list of all of the models). The number of epochs was determined by selecting the best validation performance from a model trained over 50 epochs. This was different for each model. For eICU this was 8 (LSTM), 30 (CW LSTM), 15 (Transformer) and 15 (TPC). For MIMIC-IV this was 8 (LSTM), 20 (CW LSTM), 15 (Transformer) and 10 (TPC). We noted that the best LSTM hyperparameters (Table 8) were similar to that found in Sheikhalishahi et al. [51].

All deep learning methods were implemented in PyTorch [38] and were optimised using Adam [27]. The data (including decay indicators) and the non-time series components of the models were the same as in TPC (Figure 10). We used trixi to structure our experiments and compare different hyperparameter choices [66].

The experiments were performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service (www.hpc.cam.ac.uk) funded by EPSRC Tier-2 capital grant EP/P020259/1.

Table 8: The LSTM model has 9 hyperparameters. We allowed the model to optimise a custom dropout rate for the LSTM layers. Note that batch normalisation is not applicable to the LSTM layers. The best parameters are shown as (eICU/MIMIC-IV).

LSTM Specific	Non-LSTM Specific
Hidden State (128/128)	Diag. Embedding Size* (64/-)
LSTM Dropout (0.2/0.25)	Main Dropout* (0.45/0)
No. LSTM Layers (2/1)	Final FC Layer Size* (17/36)
	Batch Normalisation* (True/True)
Global Parameters	
Batch Size (512/32)	
Learning Rate (0.00129/0.00163)	

C.1 Transformer

The Transformer is a multi-head self-attention model, originally designed for sequence-to-sequence tasks in natural language processing. It consists of both an encoder and decoder, however we only use the former. Our implementation is the same as the original encoder in Vaswani et al. [62], except that we add temporal masking to impose causality i.e. the current representation can only depend on current or earlier timepoints, and we omit the positional encodings because they were not found to be helpful. This is probably because we already have a feature to indicate the position in the time series (Section B.3).

Table 9: The Transformer model has 12 hyperparameters. We allowed the model to optimise a custom dropout rate for the Transformer layers. The positional encoding hyperparameter is binary; it determines whether or not we used the original positional encodings proposed by Vaswani et al. [62]. Note that batch normalisation is not applicable to the Transformer layers (the default implementation uses layer normalisation). The best parameters are shown as (eICU/MIMIC-IV).

Transformer Specific	Non-Transformer Specific
No. Attention Heads (2/1)	Diag. Embedding Size* (64/-)
Feedforward Size (256/64)	Main Dropout* (0.45/0)
d_{model} (16/32)	Final FC Layer Size* (17/36)
Transformer Dropout (0/0.05)	/True)
No. Transformer Layers (6/2)	
Global Parameters	
Batch Size (32/64)	
Learning Rate (0.00017/0.00129)	

D EVALUATION METRICS

The metrics we use are: mean absolute deviation (MAD), mean absolute percentage error (MAPE), mean squared error (MSE), mean squared loss error (MSLE), coefficient of determination (R^2) and

Table 10: Hyperparameter Search Ranges. We took a random sample from each range and converted to an integer if necessary. For the kernel sizes (not shown in the table) the range was dependent on the number of TPC layers selected (because large kernel sizes combined with a large number of layers can have an inappropriately wide range as the dilation factor increases per layer). In general the range of kernel sizes was around 2-5 (but it could be up to 10 for small numbers of TPC Layers).

Hyperparameter	Lower	Upper	Scale
Batch Size	4	512	\log_2
Dropout Rate (all)	0	0.5	Linear
Learning Rate	0.0001	0.01	\log_{10}
Batch Normalisation	True	False	
Positional Encoding	True	False	
Diagnosis Embedding Size	16	64	\log_2
Final FC Layer Size	16	64	\log_2
CW LSTM Hidden State Size	4	16	\log_2
Point. Channels	4	16	\log_2
Temp. Channels	4	16	\log_2
Temp. Channels (weight sharing)	16	64	\log_2
LSTM Hidden State Size	16	256	\log_2
d_{model}	16	256	\log_2
Feedforward Size	16	256	\log_2
No. Attention Heads	2	16	\log_2
No. TPC Layers	1	12	Linear
No. LSTM Layers	1	4	Linear
No. Transformer Layers	1	10	Linear

Cohen Kappa Score. We modify the MAPE metric slightly so that very small true LoS values do not produce unbounded MAPE values. We place a 4 hour lower bound on the divisor i.e.

$$\text{Absolute Percentage Error} = \left| \frac{y_{true} - y_{pred}}{\max(y_{true}, \frac{4}{24})} \right| * 100$$

MAD and MAPE are improved by centering predictions on the median. Likewise, MSE and R^2 are bettered by centering predictions around the mean. They are more affected by the skew. MSLE is a good metric for this task, indeed, it is the loss function in most experiments, but is less readily-interpretable than some of the other measures. Cohen's linear weighted Kappa Score [7] is intended for ordered classification tasks rather than regression, but it can effectively mitigate for skew if the bins are chosen well. It has previously provided useful insights in Harutyunyan et al. [18], so we use the same LoS bins: 0-1, 1-2, 2-3, 3-4, 4-5, 5-6, 6-7, 7-8, 8-14, and 14+ days. As a classification measure, it will treat everything falling within the same classification bin as equal, so it is fundamentally a coarser measure than the other metrics.

To illustrate the importance of using multiple metrics, consider that the mean and median models are in some sense equally poor (neither has learned anything meaningful for our purposes). Nevertheless, the median model is able to better exploit the MAD, MAPE and MSLE metrics, and the mean model fares better with MSE, but the Kappa score betrays them both. A good model will perform well across all of the metrics.

Table 11: Performance of the TPC model and its baselines when only some of the time series are included (the flat features and diagnoses are still included). The indicator ‘(labs)’ means that only the laboratory tests were included, ‘(other)’ refers to everything except labs: vital signs, nurse observations and machine logged variables. The metric acronyms, colour scheme and confidence interval calculations are described in Table 2. The percentage impairment when compared to the complete dataset is shown in grey underneath the absolute values. They are calculated with respect to the best value for the metric: 0 for MAD, MAPE, MSE and MSLE, and 1 for R^2 and Kappa. A large percentage impairment means that the model does much better with complete data i.e. it has a high ‘percentage gain’ from the combination of both data types compared to the ablation case.

Model	MAD	MAPE	MSE	MSLE	R^2	Kappa
LSTM	2.39±0.00	118.2±1.1	26.9±0.1	1.47±0.01	0.09±0.00	0.28±0.00
LSTM (labs)	2.43±0.00 (-1.7%)	123.8±1.2 (-4.7%)	27.3±0.1 (-1.5%)	1.57±0.00 (-6.8%)	0.08±0.00 (-1.1%)	0.27±0.00 (-1.4%)
LSTM (other)	2.41±0.00 (-0.8%)	120.2±0.7 (-1.7%)	27.3±0.1 (-1.5%)	1.49±0.00 (-1.4%)	0.07±0.00 (-2.2%)	0.27±0.00 (-1.4%)
CW LSTM	2.37±0.00	114.5±0.4	26.6±0.1	1.43±0.00	0.10±0.00	0.30±0.00
CW LSTM (labs)	2.42±0.00 (-2.1%)	124.4±0.7 (-8.6%)	27.0±0.1 (-1.5%)	1.57±0.00 (-9.8%)	0.08±0.00 (-2.2%)	0.28±0.00 (-2.9%)
CW LSTM (other)	2.41±0.00 (-1.7%)	120.6±0.8 (-5.3%)	27.1±0.1 (-1.9%)	1.51±0.00 (-5.6%)	0.08±0.00 (-2.2%)	0.29±0.00 (-1.4%)
Transformer	2.36±0.00	114.1±0.6	26.7±0.1	1.43±0.00	0.09±0.00	0.30±0.00
Transformer (labs)	2.42±0.00 (-2.5%)	121.0±0.7 (-6.0%)	27.3±0.1 (-2.2%)	1.56±0.00 (-9.1%)	0.07±0.00 (-2.2%)	0.27±0.00 (-4.3%)
Transformer (other)	2.40±0.00 (-1.7%)	118.3±0.6 (-3.7%)	27.3±0.1 (-2.2%)	1.50±0.00 (-4.9%)	0.07±0.00 (-2.2%)	0.27±0.00 (-4.3%)
TPC	1.78±0.02	63.5±4.3	21.7±0.5	0.70±0.03	0.27±0.02	0.58±0.01
TPC (labs)	1.85±0.01 (-3.9%)	72.0±2.2 (-13.4%)	22.5±0.2 (-3.7%)	0.81±0.01 (-15.7%)	0.24±0.01 (-4.1%)	0.55±0.00 (-7.1%)
TPC (other)	1.81±0.02 (-1.7%)	68.5±4.7 (-7.9%)	21.8±0.3 (-0.5%)	0.77±0.03 (-10.0%)	0.26±0.01 (-1.4%)	0.57±0.01 (-2.4%)

E TIME SERIES ABLATION

We performed ablations on the type of time series variable that we include: laboratory tests only (labs), which are infrequently sampled, and all other variables (other) which include vital signs, nurse observations, and automatically recorded variables (e.g. from ventilator machines). This shows how well each model can cope with time series exhibiting different periodicity and sampling frequencies. The results are shown in Table 11. The TPC model has the largest percentage gain when the labs and other variables are combined (this is synonymous with the greatest percentage impairment in the ablations). Next are the CW LSTM and Transformer, followed by the LSTM. This suggests that the TPC model is best able to exploit EHR time series with different temporal properties.

When examining the results for LSTM and CW LSTM in more detail, we can see that the CW LSTM only has an advantage when the model has to combine the data types. This supports the hypothesis that the CW LSTM is better able to cope when there are varying frequencies in the data, as it can tailor the processing to each. When the inter-feature variability is small (the same type of time series) they perform similarly.

It is unsurprising that the Transformer does better than the LSTM when combining data types, as it can directly skip over large gaps in time to extract a trend in lab values, while simultaneously attending to recent timepoints for the processing of other variables.

The TPC is the most successful model; its inherent periodic structure helps it to extract useful information from all of the variables. The CW LSTM and Transformer do not have this in their architectures, making the derivation more obscure. The importance of periodicity is discussed in more detail in Section 8.

Table 12: The effect of changing the size of the training data on the LSTM, CW LSTM, Transformer, and TPC model performance in the eICU dataset. A hundred percent of the training set represents 102,712 ICU stays, 50% is 51,356, 25% is 25,678, 12.5% is 12,839, and 6.25% is 6,420 stays.

Model (% train data)	MAD	MAPE	MSE	MSLE	R^2	Kappa
LSTM (100)	2.39±0.00	118.2±1.1	26.9±0.1	1.47±0.01	0.09±0.00	0.28±0.00
LSTM (50)	2.41±0.01	129.9±1.9	26.2±0.2	1.52±0.00	0.11±0.01	0.31±0.01
LSTM (25)	2.44±0.01	126.8±2.5	27.2±0.3	1.58±0.00	0.08±0.01	0.27±0.01
LSTM (12.5)	2.48±0.01	137.4±3.4	27.4±0.2	1.65±0.01	0.07±0.01	0.27±0.01
LSTM (6.25)	2.52±0.02	135.9±3.3	28.0±0.8	1.71±0.02	0.05±0.03	0.26±0.03
CW LSTM (100)	2.37±0.00	114.5±0.4	26.6±0.1	1.43±0.00	0.10±0.00	0.30±0.00
CW LSTM (50)	2.40±0.01	123.4±0.7	26.5±0.1	1.48±0.01	0.10±0.00	0.31±0.00
CW LSTM (25)	2.44±0.00	119.8±1.3	27.2±0.1	1.54±0.00	0.08±0.00	0.29±0.00
CW LSTM (12.5)	2.50±0.01	134.7±1.5	27.7±0.1	1.63±0.01	0.06±0.00	0.28±0.00
CW LSTM (6.25)	2.58±0.01	129.8±3.5	29.0±0.2	1.73±0.01	0.02±0.01	0.25±0.01
Transformer (100)	2.36±0.00	114.1±0.6	26.7±0.1	1.43±0.00	0.09±0.00	0.30±0.00
Transformer (50)	2.39±0.00	120.1±0.6	26.5±0.1	1.48±0.00	0.10±0.00	0.31±0.00
Transformer (25)	2.43±0.01	117.9±1.8	27.2±0.2	1.54±0.01	0.08±0.01	0.28±0.01
Transformer (12.5)	2.48±0.01	128.1±2.3	27.9±0.1	1.62±0.01	0.06±0.00	0.26±0.01
Transformer (6.25)	2.52±0.01	139.7±2.4	27.8±0.1	1.69±0.02	0.06±0.00	0.26±0.00
TPC (100)	1.78±0.02	63.5±4.3	21.7±0.5	0.70±0.03	0.27±0.02	0.58±0.01
TPC (50)	1.95±0.02	72.0±3.1	23.8±0.4	0.87±0.03	0.19±0.01	0.51±0.01
TPC (25)	2.09±0.01	89.0±3.8	24.8±0.3	1.09±0.02	0.16±0.01	0.45±0.01
TPC (12.5)	2.28±0.01	101.4±4.8	27.0±0.4	1.36±0.03	0.08±0.01	0.35±0.02
TPC (6.25)	2.49±0.02	139.9±5.5	28.0±0.3	1.64±0.03	0.05±0.01	0.28±0.01

Table 13: The effect of training with the mean squared logarithmic error (MSLE) loss function when compared to mean squared error (MSE) on the eICU dataset. This is an extension to Table 3 (refer to its legend for definitions of the metric acronyms, detailed of CI calculations and meaning of the colour scheme).

Model	MAD	MAPE	MSE	MSLE	R^2	Kappa
LSTM (MSE)	2.57±0.03	235.2±6.2	24.5±0.2**	1.97±0.02	0.17±0.01**	0.28±0.01
LSTM (MSLE)	2.39±0.00**	118.2±1.1**	26.9±0.1	1.47±0.01**	0.09±0.00	0.28±0.00
CW LSTM (MSE)	2.56±0.01	218.5±4.0	24.2±0.1**	1.84±0.02	0.18±0.00**	0.34±0.01**
CW LSTM (MSLE)	2.37±0.00**	114.5±0.4**	26.6±0.1	1.43±0.00**	0.10±0.00	0.30±0.00
Transformer (MSE)	2.51±0.01	212.7±5.2	24.7±0.2**	1.87±0.03	0.16±0.01**	0.28±0.01
Transformer (MSLE)	2.36±0.00**	114.1±0.6**	26.7±0.1	1.43±0.00**	0.09±0.00	0.30±0.00**
TPC (MSE)	2.21±0.02	154.3±10.1	21.6±0.2	1.80±0.10	0.27±0.01	0.47±0.01
TPC (MSLE)	1.78±0.02**	63.5±4.3**	21.7±0.5	0.70±0.03**	0.27±0.02	0.58±0.01**

Table 14: eICU multitask results. We compare the performance of each model on individual tasks (LoS or mortality prediction) to the multitask setting (both LoS and mortality). The results from Table 2a are repeated here for ease of comparison. Note that the ‘mean’ and ‘median’ models are only for LoS – there is no equivalent model for mortality prediction.

Model	In-Hospital Mortality		Length of Stay					
	AUROC	AUPRC	MAD	MAPE	MSE	MSLE	R^2	Kappa
Mean	–	–	3.21	395.7	29.5	2.87	0.00	0.00
Median	–	–	2.76	184.4	32.6	2.15	-0.11	0.00
LSTM	0.849±0.002	0.407±0.012	–	–	–	–	–	–
	–	–	2.39±0.00	118.2±1.1	26.9±0.1*	1.47±0.01	0.09±0.00*	0.28±0.00
	0.852±0.003	0.436±0.007**	2.40±0.01	116.5±0.8*	27.2±0.2	1.47±0.01	0.08±0.01	0.28±0.01
CW LSTM	0.855±0.001	0.464±0.004	–	–	–	–	–	–
	–	–	2.37±0.00	114.5±0.4	26.6±0.1*	1.43±0.00*	0.10±0.00*	0.30±0.00
	0.865±0.002**	0.490±0.007**	2.37±0.00	115.0±0.7	26.8±0.1	1.44±0.00	0.09±0.00	0.30±0.00
Transformer	0.851±0.002	0.454±0.005	–	–	–	–	–	–
	–	–	2.36±0.00	114.1±0.6	26.7±0.1	1.43±0.00	0.09±0.00	0.30±0.00
	0.858±0.001**	0.475±0.004**	2.36±0.00	114.2±0.7	26.6±0.1	1.43±0.00	0.10±0.00	0.30±0.00
TPC	0.864±0.001	0.508±0.005	–	–	–	–	–	–
	–	–	1.78±0.02	63.5±3.8	21.8±0.5	0.71±0.03	0.26±0.02	0.58±0.01
	0.865±0.002	0.523±0.006**	1.55±0.01**	46.4±2.6**	18.7±0.2**	0.40±0.02**	0.37±0.01**	0.70±0.00**

Table 15: MIMIC-IV multitask results.

Model	In-Hospital Mortality		Length of Stay					
	AUROC	AUPRC	MAD	MAPE	MSE	MSLE	R^2	Kappa
Mean	–	–	5.24	474.9	77.7	2.80	0.00	0.00
Median	–	–	4.60	216.8	86.8	2.09	-0.12	0.00
LSTM	0.895±0.001	0.657±0.003	–	–	–	–	–	–
	–	–	3.68±0.02	107.2±3.1	65.7±0.7	1.26±0.01	0.15±0.01	0.43±0.01
	0.896±0.002	0.659±0.004	3.66±0.01	106.8±2.7	65.3±0.6	1.25±0.01*	0.16±0.01	0.44±0.00
CW LSTM	0.897±0.002	0.650±0.005	–	–	–	–	–	–
	–	–	3.68±0.02	107.0±1.8	66.4±0.6	1.23±0.01	0.15±0.01	0.43±0.00
	0.899±0.002	0.654±0.003	3.69±0.02	107.2±1.6	66.3±0.6	1.23±0.01	0.15±0.01	0.44±0.00
Transformer	0.890±0.002	0.641±0.008	–	–	–	–	–	–
	–	–	3.62±0.02	113.8±1.8	63.4±0.5	1.21±0.01	0.18±0.01	0.45±0.00
	0.898±0.001**	0.656±0.005*	3.61±0.01	112.3±2.0	63.3±0.3	1.20±0.01	0.19±0.00	0.45±0.00
TPC	0.905±0.001	0.691±0.006	–	–	–	–	–	–
	–	–	2.39±0.03	47.6±1.4	46.3±1.3	0.39±0.02	0.40±0.02	0.78±0.01
	0.918±0.002**	0.713±0.007**	2.28±0.07*	32.4±1.2**	42.0±1.2**	0.19±0.00**	0.46±0.02**	0.85±0.00**

Table 16: eICU time series features. ‘Time in the ICU’ and ‘Time of day’ were not part of the tables in eICU but were added later as helpful indicators to the model.

Source Table			
<i>lab</i>			<i>respiratorycharting</i>
-basos	MPV	glucose	Exhaled MV
-eos	O2 Sat (%)	lactate	Exhaled TV (patient)
-lymphs	PT	magnesium	LPM O2
-monos	PT - INR	pH	Mean Airway Pressure
-polys	PTT	paCO2	Peak Insp. Pressure
ALT (SGPT)	RBC	paO2	PEEP
AST (SGOT)	RDW	phosphate	Plateau Pressure
BUN	WBC x 1000	platelets x 1000	Pressure Support
Base Excess	albumin	potassium	RR (patient)
FiO2	alkaline phos.	sodium	SaO2
HCO3	anion gap	total bilirubin	TV/kg IBW
Hct	bedside glucose	total protein	Tidal Volume (set)
Hgb	bicarbonate	troponin - I	Total RR
MCH	calcium	urinary specific gravity	Vent Rate
MCHC	chloride		
MCV	creatinine		
<i>nursecharting</i>	<i>vitalperiodic</i>	<i>vitalaperiodic</i>	N/A
Bedside Glucose	cvp	noninvasivediastolic	Time in the ICU
Delirium Scale/Score	heartrate	noninvasivemean	Time of day
Glasgow coma score	respiration	noninvasivesystolic	
Heart Rate	sao2		
Invasive BP	st1		
Non-Invasive BP	st2		
O2 Admin Device	st3		
O2 L/%	systemicdiastolic		
O2 Saturation	systemicmean		
Pain Score/Goal	systemicsystolic		
Respiratory Rate	temperature		
Sedation Score/Goal			
Temperature			

Table 17: MIMIC-IV time series features.

Source Table		
<i>chartevents</i>		
Activity / Mobility (JH-HLM)	Mean Airway Pressure	Resp Alarm - High
Apnea Interval	Minute Volume	Resp Alarm - Low
Arterial Blood Pressure Alarm - High	Minute Volume Alarm - High	Respiratory Rate
Arterial Blood Pressure Alarm - Low	Minute Volume Alarm - Low	Respiratory Rate (Set)
Arterial Blood Pressure diastolic	Non Invasive Blood Pressure diastolic	Respiratory Rate (Total)
Arterial Blood Pressure mean	Non Invasive Blood Pressure mean	Respiratory Rate (spontaneous)
Arterial Blood Pressure systolic	Non Invasive Blood Pressure systolic	Richmond-RAS Scale
Braden Score	Non-Invasive Blood Pressure Alarm - High	Strength L Arm
Current Dyspnea Assessment	Non-Invasive Blood Pressure Alarm - Low	Strength L Leg
Daily Weight	O2 Flow	Strength R Arm
Expiratory Ratio	O2 Saturation Pulseoxymetry Alarm - Low	Strength R Leg
Fspn High	O2 saturation pulseoxymetry	Temperature Fahrenheit
GCS - Eye Opening	PEEP set	Tidal Volume (observed)
GCS - Motor Response	PSV Level	Tidal Volume (set)
GCS - Verbal Response	Pain Level	Tidal Volume (spontaneous)
Glucose finger stick (range 70-100)	Pain Level Response	Total PEEP Level
Heart Rate	Paw High	Ventilator Mode
Heart Rate Alarm - Low	Peak Insp. Pressure	Vti High
Heart rate Alarm - High	Phosphorous	
Inspired O2 Fraction	Plateau Pressure	
<i>labevents</i>		N/A
Alanine Aminotransferase (ALT)	MCHC	Time in the ICU
Alkaline Phosphatase	MCV	Time of day
Anion Gap	Magnesium	
Asparate Aminotransferase (AST)	Oxygen Saturation	
Base Excess	PT	
Bicarbonate	PTT	
Bilirubin, Total	Phosphate	
Calcium, Total	Platelet Count	
Calculated Total CO2	Potassium	
Chloride	Potassium, Whole Blood	
Creatinine	RDW	
Free Calcium	RDW-SD	
Glucose	Red Blood Cells	
H	Sodium	
Hematocrit	Sodium, Whole Blood	
Hematocrit, Calculated	Temperature	
Hemoglobin	Urea Nitrogen	
I	White Blood Cells	
INR(PT)	pCO2	
L	pH	
Lactate	pO2	
MCH		