Using Features from Pre-trained TimeNet for Clinical Predictions

Conference Paper · July 2018

CITATIONS
21

READS
1,165

4 authors:

Priyanka Gupta
Tata Consultancy Services Limited
12 PUBLICATIONS **228** CITATIONS
SEE PROFILE

Pankaj Malhotra
Tata Consultancy Services Limited
47 PUBLICATIONS **2,840** CITATIONS
SEE PROFILE

Lovekesh Vig
Tata Consultancy Services Limited
169 PUBLICATIONS **4,638** CITATIONS
SEE PROFILE

Gautam Shroff
Tata Consultancy Services Limited
155 PUBLICATIONS **3,914** CITATIONS
SEE PROFILE

Using Features from Pre-trained TimeNet for Clinical Predictions

Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, Gautam Shroff
TCS Research, New Delhi, India
{priyanka.g35, malhotra.pankaj, lovekesh.vig, gautam.shroff}@tcs.com,

Abstract

Predictive models based on Recurrent Neural Networks (RNNs) for clinical time series have been successfully used for various tasks such as phenotyping, in-hospital mortality prediction, and diagnostics. However, RNNs require large labeled data for training and are computationally expensive to train. Pre-training a network for some supervised or unsupervised tasks on a dataset, and then fine-tuning via transfer learning for a related end-task can be an efficient way to leverage deep models for scenarios that lack in either computational resources or labeled data, or both. In this work, we consider an approach to leverage a deep RNN – namely *TimeNet* [Malhotra *et al.*, 2017] – that is pre-trained on a large number of diverse publicly available time-series from UCR Repository [Chen *et al.*, 2015]. TimeNet maps varying-length time series to fixed-dimensional feature vectors and acts as an off-the-shelf feature extractor. TimeNet-based approach overcome the need for hand-crafted features, and allows for use of traditional easy-to-train and interpretable linear models for the end-task, while still leveraging the features from a deep neural network. Empirical evaluation of the proposed approach on MIMIC-III[1] data suggests promising direction for future exploration: our results are comparable to existing benchmarks while our models require lesser training and hyper-parameter tuning effort.

1 Introduction

There has been a growing interest in using deep learning models for various clinical prediction tasks from Electronic Health Records, e.g. Doctor AI [Choi *et al.*, 2016] for medical diagnosis, Deep Patient [Miotto *et al.*, 2016] to predict future diseases in patients, DeepR [Nguyen *et al.*, 2017] to predict unplanned readmission after discharge, etc. With various medical parameters being recorded over a period of time in EHR databases, Recurrent Neural Networks (RNNs) can be an effective way to model the sequential aspects of EHR data, e.g. diagnoses [Lipton *et al.*, 2015; Che *et al.*, 2016; Choi *et al.*, 2016], mortality prediction and estimating length of stay [Harutyunyan *et al.*, 2017; Purushotham *et al.*, 2017; Rajkomar *et al.*, 2018].

However, training RNNs requires large labeled training data like any other deep learning approach, and can be computationally inefficient because of sequential nature of computations. On the other hand, training a deep network on diverse instances can provide generic features for unseen instances, e.g. VGGNet [Simonyan and Zisserman, 2014] for images. Also, fine-tuning a pre-trained network with transfer learning is often faster and easier than constructing and training a new network from scratch [Bengio, 2012]. The advantage of learning in such a manner is that the pre-trained network has already learned a rich set of features that can then be applied to a wide range of other similar tasks.

Deep RNNs have been shown to perform hierarchical processing of time series with different layers tackling different time scales [Hermans and Schrauwen, 2013; Malhotra *et al.*, 2015]. TimeNet [Malhotra *et al.*, 2017] is a general-purpose multi-layered RNN trained on large number of diverse time series from UCR Time Series Archive [Chen *et al.*, 2015] (refer Section 3 for details) that has been shown to be useful as off-the-shelf feature extractor for time series. TimeNet has been trained on 18 different datasets simultaneously via an RNN autoencoder in an unsupervised manner for reconstruction task. Features extracted from TimeNet have been found to be useful for classification task on 25 datasets not seen during training of TimeNet, proving its ability to provide meaningful features for unseen datasets.

In this work, *we provide an efficient way to learn prediction models for clinical time series by leveraging general-purpose features via TimeNet*. TimeNet maps variable-length clinical time series to fixed-dimensional feature vectors, that are subsequently used for patient phenotyping and in-hospital mortality prediction tasks on MIMIC-III database [Johnson *et al.*, 2016] via easily trainable non-temporal linear classification models. We observe that TimeNet-based features can be used to build such classification models with very little train-

[1] TimeNet-based features for MIMIC-III time series are available on request from authors.

Presented at The 3rd International Workshop on Knowledge Discovery in Healthcare Data at IJCAI 2018, Stockholm, Sweden.

ing effort while yielding performance comparable to models with hand-crafted features or carefully trained domain-specific RNNs, as benchmarked in [Harutyunyan *et al.*, 2017; Song *et al.*, 2017]. Further, we propose a simple mechanism to leverage the weights of the linear classification models to provide insights into the relevance of each raw input feature (physiological parameter) for a given phenotype (discussed in Section 4.2).

2 Related Work

TimeNet-based features have been shown to be useful for various tasks including ECG classification [Malhotra *et al.*, 2017]. In this work, we consider application of TimeNet to phenotyping and in-hospital mortality tasks for multivariate clinical time series classification. Deep Patient [Miotto *et al.*, 2016] proposes leveraging features from a pre-trained stacked-autoencoder for EHR data. However, it does not leverage the temporal aspect of the data and uses a non-temporal model based on stacked-autoencoders. Our approach extracts temporal features via TimeNet incorporating the sequential nature of EHR data. Doctor AI [Choi *et al.*, 2016] uses discretized medical codes (e.g. diagnosis, medication, procedure) from longitudinal patient visits via a purely supervised setting while we use real-valued time series. While approaches like Doctor AI require training a deep RNN from scratch, our approach leverages a general-purpose RNN for feature extraction.

[Harutyunyan *et al.*, 2017] consider training a deep RNN model for multiple prediction tasks simultaneously including phenotyping and in-hospital mortality to learn a general-purpose deep RNN for clinical time series. They show that it is possible to train a single network for multiple tasks simultaneously by capturing generic features that work across different tasks. We also consider leveraging generic features for clinical time series but using an RNN that is pre-trained on diverse time series across domains, making our approach more efficient. Further, we provide an approach to rank the raw input features in order of their relevance that helps validate the models learned.

3 Background: TimeNet

TimeNet [Malhotra *et al.*, 2017] is a pre-trained off-the-shelf feature extractor for univariate time series with three recurrent layers having 60 Gated Recurrent Units (GRUs) [Cho *et al.*, 2014] each. TimeNet is an RNN trained via an autoencoder consisting of an encoder RNN and a decoder RNN trained simultaneously using the sequence-to-sequence learning framework [Sutskever *et al.*, 2014; Bahdanau *et al.*, 2014] as shown in Figure 1(a). RNN autoencoder is trained to obtain the parameters $\mathbf{W}_E$ of the encoder RNN f_E via reconstruction task such that for input $x_{1...T} = x_1, x_2, ..., x_T$ ($x_i \in \mathbb{R}$), the target output time series $x_{T...1} = x_T, x_{T-1}, ..., x_1$ is reverse of the input.

The RNN encoder f_E provides a non-linear mapping of the univariate input time series to a fixed-dimensional vector representation $\mathbf{z}_T$: $\mathbf{z}_T = f_E(x_{1...T}; \mathbf{W}_E)$, followed by an RNN decoder f_D based non-linear mapping of $\mathbf{z}_T$ to univariate time series: $\hat{x}_{T...1} = f_D(\mathbf{z}_T; \mathbf{W}_D)$; where $\mathbf{W}_E$ and $\mathbf{W}_D$ are the parameters of the encoder and decoder, respectively. The model is trained to minimize the average squared reconstruction error. Training on 18 diverse datasets simultaneously results in robust time series features getting captured in $\mathbf{z}_T$: the decoder relies on $\mathbf{z}_T$ as the only input to reconstruct the time series, forcing the encoder to capture all the relevant information in the time series into the fixed-dimensional vector $\mathbf{z}_T$. This vector $\mathbf{z}_T$ is used as the feature vector for input $x_{1...T}$. This feature vector is then used to train a simpler classifier (e.g. SVM, as used in [Malhotra *et al.*, 2017]) for the end task. TimeNet maps a univariate input time series to 180-dimensional feature vector, where each dimension corresponds to final output of one of the 60 GRUs in the 3 recurrent layers.

4 TimeNet Features for Clinical Time Series

Consider a set $\mathcal{D}$ of labeled time series instances from an EHR database: $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, where $\mathbf{x}^{(i)}$ is a multivariate time series, $y^{(i)} \in \{y_1, \ldots, y_C\}$, C is the number of classes, N is the number of unique patients (in our experiments, we consider each episode of hospital stay for a patient as a separate data instance). In this work, we consider presence or absence of a phenotype as a binary classification task such that $C = 2$. We learn an independent model for each phenotype (unlike [Harutyunyan *et al.*, 2017] which consider phenotyping as a multi-label classification problem). This allows us to build simple linear binary classification models as described next in Section 4.1. In practice, the outputs of these binary classifiers can then be considered together to estimate the set of phenotypes present in a patient. Similarly, mortality prediction is considered to be a binary classification task where the goal is to classify whether the patient will survive (after admission to ICU) or not.

4.1 Classification using TimeNet features

Feature Extraction for Multivariate Clinical Time Series

For a multivariate time series $\mathbf{x} = \mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_T$, where $\mathbf{x}_t \in \mathbb{R}^n$, we consider time series for each of the n raw input features (physiological parameters, e.g. glucose level, heart rate, etc.) independently, to obtain univariate time series $x_j = x_{j1}x_{j2} \ldots x_{jT}, j = 1 \ldots n$. (Note: We use $\mathbf{x}$ instead of $\mathbf{x}^{(i)}$ and omit superscript (i) for ease of notation). We obtain the vector representation $\mathbf{z}_{jT} = f_E(x_j; \mathbf{W}_E)$ for x_j, where $\mathbf{z}_{jT} \in \mathbb{R}^c$ using TimeNet as f_E with $c = 180$ (as described in Section 3). In general, time series length T also depends on i, e.g. based on length of stay in hospital. We omit this for sake of clarity without loss of generality. In practice, we convert each time series to have equal length T by suitable pre/post-padding with 0s. We concatenate the TimeNet-features $\mathbf{z}_{jT}$ for each raw input feature j to get the final feature vector $\mathbf{z}_T = [\mathbf{z}_{1T}, \mathbf{z}_{2T}, \ldots, \mathbf{z}_{nT}]$ for time series $\mathbf{x}$, where $\mathbf{z}_T \in \mathbb{R}^m$, $m = n \times c$ as illustrated in Figure 1(b).

Using TimeNet-based Features for Classification

The final concatenated feature vector $\mathbf{z}_T$ is used as input for the phenotyping and mortality prediction classification tasks. We note that since $c = 180$ is large, $\mathbf{z}_T$ has large number of features $m \geq 180$. We consider a linear mapping from input

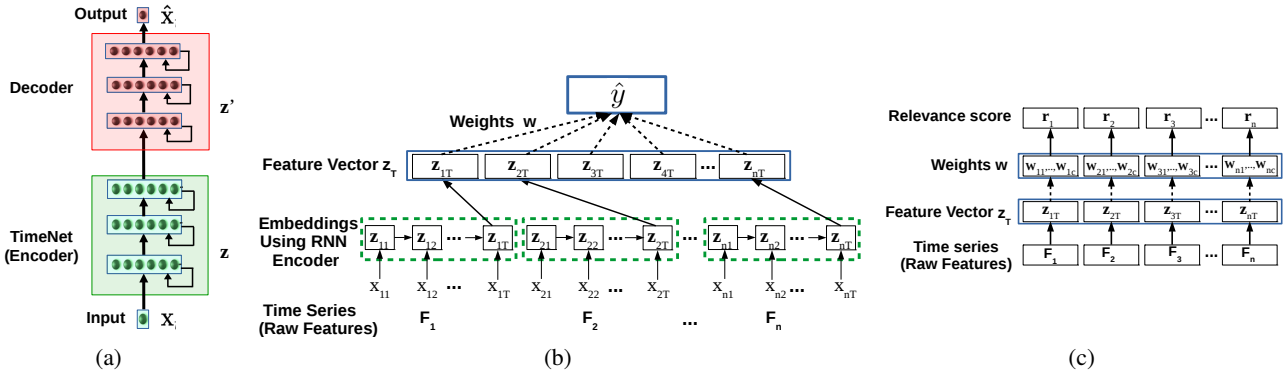


Figure 1: (a) TimeNet trained via RNN Encoder-Decoder with three hidden GRU layers. (b) TimeNet based Feature Extraction. TimeNet is shown unrolled over time. (c) Obtaining relevance scores for raw input features. Here, T: time series length, n: number of raw input features.

TimeNet features $\mathbf{z}_T$ to the target label y s.t. the estimate $\hat{y} = \mathbf{w} \cdot \mathbf{z}_T$, where $\mathbf{w} \in \mathbb{R}^m$. We constrain the linear model with weights $\mathbf{w}$ to use only a few of these large number of features. The weights are obtained using LASSO-regularized loss function [Tibshirani, 1996]:

$$\arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \mathbf{w} \cdot \mathbf{z}_T^{(i)})^2 + \alpha ||\mathbf{w}||_1 \qquad (1)$$

where $y^{(i)} \in \{0, 1\}$, $||\mathbf{w}||_1 = \sum_{j=1}^{n} \sum_{k=1}^{c} |w_{jk}|$ is the L_1-norm, where w_{jk} represents the weight assigned to the k-th TimeNet-feature for the j-th raw feature, and α controls the extent of sparsity – with higher α implying more sparsity, i.e. fewer TimeNet features are selected for the final classifier.

4.2 Obtaining Relevance Scores for Raw Features

Determining relevance of the n raw input features for a given phenotype is potentially useful to obtain insights into the obtained classification model. The sparse weights $\mathbf{w}$ are easy to interpret and can give interesting insights into relevant features for a classification task (e.g. as used in [Micenková *et al.*, 2013]). We obtain the relevance r_j of the j-th raw input feature as the sum of the absolute values of the weights w_{jk} assigned to the corresponding TimeNet features $\mathbf{z}_{jT}$ as shown in Figure 1(c), s.t.

$$r_j = \sum_{k=1}^{c} |w_{jk}|, \; j = 1 \ldots n. \qquad (2)$$

Further, r_j is normalized using min-max normalization such that $r'_j = \frac{r_j - r_{min}}{r_{max} - r_{min}} \in [0, 1]$; r_{min} is minimum of $\{r_1, \ldots, r_n\}$, r_{max} is maximum of $\{r_1, \ldots, r_n\}$. In practice, this kind of relevance scores for the raw features help to interpret and validate the overall model. For example, one would expect blood glucose level feature to have a high relevance score when learning a model to detect diabetes mellitus phenotype (we provide such insights later in Section 5).

5 Experimental Evaluation

5.1 Dataset Details

We use MIMIC-III (v1.4) clinical database [Johnson *et al.*, 2016] which consists of over 60,000 ICU stays across 40,000 critical care patients. We use same experimental setup as in [Harutyunyan *et al.*, 2017], with same splits and features for train, validation and test datasets[2] based on 17 physiological time series with 12 real-valued and 5 categorical time series, sampled at 1 hour intervals. The categorical variables are converted to one-hot vectors such that final multivariate time series has $n = 76$ raw input features (59 actual features and 17 masking features to denote missing values).

For phenotyping task, the goal is to classify 25 phenotypes common in adult ICUs. For in-hospital mortality task, the goal is to predict whether the patient will survive or not given the time series observations up to 48 hours. In all our experiments, we restrict training time series data up to first 48 hours in ICU stay, such that $T = 48$ while training all models to imitate practical scenario where early predictions are important, unlike [Harutyunyan *et al.*, 2017; Song *et al.*, 2017] which use entire time series for training the classifier for phenotyping task.

5.2 Evaluation

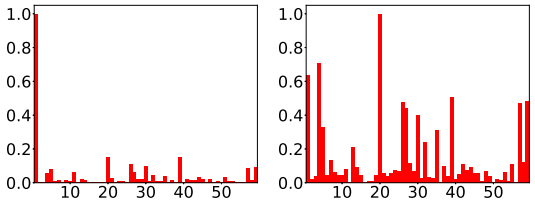
We have $n = 76$ raw input features resulting in $m = 13,680$-dimensional ($m = 76 \times 180$) TimeNet feature vector for each admission. We use $\alpha = 0.0001$ for phenotype classifiers and use $\alpha = 0.0003$ for in-hospital mortality classifier (α is chosen based on hold-out validation set). Table 1 summarizes the results and provides comparison with existing benchmarks. Refer Table 2 for detailed phenotype-wise results.

We consider two variants of classifier models for phenotyping task: i) *TimeNet-x* using data from current episode, ii) *TimeNet-x-Eps* using data from previous episode of a patient as well (whenever available) via an additional input feature related to presence or absence of the phenotype in previous episode. Each classifier is trained using up to first 48 hours of data after ICU admission. However, we consider two classifier variants depending upon hours of data x used to estimate the target class at test time. For $x = 48$, data up to first 48 hours after admission is used for determining the phenotype. For $x = All$, the learned classifier is applied to all 48-hours windows (overlapping with shift of 24 hours) over the entire ICU stay period of a patient, and the average phenotype

[2] https://github.com/yerevann/mimic3-benchmarks

Table 1: Classification Performance Comparison. Here, LR: Logistic regression, LSTM-Multi: LSTM-based multitask model, SAnD (Simply Attend and Diagnose): Fully attention-based model, SAnD-Multi: SAnD-based multitask model. (Note: *For phenotyping, we compare TimeNet-48-Eps with existing benchmarks over TimeNet-All-Eps as it is more applicable in practical scenarios. **Only TimeNet-48 variant is applicable for in-hospital mortality task.)

	[Harutyunyan *et al.*, 2017]			[Song *et al.*, 2017]		Proposed (Features using [Malhotra *et al.*, 2017])			
Metric	**LR**	**LSTM**	**LSTM-Multi**	**SAnD**	**SAnD-Multi**	**TimeNet-48**	**TimeNet-All**	**TimeNet-48-Eps**	**TimeNet-All-Eps***
Task 1: Phenotyping									
Micro AUC	0.801	**0.821**	0.817	0.816	0.819	0.812	0.813	0.820	*0.822*
Macro AUC	0.741	0.77	0.766	0.766	0.771	0.761	0.764	**0.772**	*0.775*
Weighted AUC	0.732	0.757	0.753	0.754	0.759	0.751	0.754	**0.765**	*0.768*
Task 2: In-Hospital Mortality Prediction**									
AUROC	0.845	0.854	**0.863**	0.857	0.859	0.852	-	-	-
AUPRC	0.472	0.516	0.517	0.518	**0.519**	**0.519**	-	-	-
min(Se,$^{+}$P)	0.469	0.491	0.499	0.5	**0.504**	0.486	-	-	-



(a) P_1 (b) P_2

Figure 2: Feature relevance after LASSO. x-axis: Feature Number, y-axis: Relevance Score. Here, P_1: Diabetes Mellitus with Complications, P_2: Essential Hypertension.

probability across windows is used as the final estimate of the target class. In *TimeNet-x-Eps*, the additional feature is related to the presence (1) or absence (0) of the phenotype during the previous episode. We use the ground-truth value for this feature during training time, and the probability of presence of phenotype during previous episode (as given via LASSO-based classifier) at test time.

5.3 Observations

Classification Tasks

For the phenotyping task, we make following observations from Table 1:

1. *TimeNet-48 vs LR*: *TimeNet-based features perform significantly better than hand-crafted features* as used in LR (logistic regression), while using first 48 hours of data only unlike the LR approach that uses entire episode's data. This proves the effectiveness of TimeNet features for MIMIC-III data. Further, it only requires *tuning a single hyperparameter* α for LASSO, unlike other approaches like LSTM [Harutyunyan *et al.*, 2017] that would involve tuning number of hidden units, layers, learning rate, etc.
2. *TimeNet-x vs TimeNet-x-Eps*: Leveraging previous episode's time series data for a patient significantly improves the classification performance.
3. *TimeNet-48-Eps* performs better than existing benchmarks, while still being *practically more feasible* as it looks at only up to 48 hours of current episode of a patient rather than the entire current episode. For in-hospital mortality task, we observe comparable performance to existing benchmarks.

Training linear models is significantly fast and it took around 30 minutes for obtaining any of the binary classifiers while tuning for $\alpha \in [10^{-5} - 10^{-3}]$ (five equally-spaced values) on a 32GB RAM machine with Quad Core i7 2.7GHz processor.

We observe that LASSO leads to 96.2 ± 0.8 % sparsity (i.e. percentage of weights $w_{jk} \approx 0$) for all classifiers leading to around 550 useful features (out of 13,680) for each phenotype classification.

Relevance Scores for Raw Input Features

We observe intuitive interpretation for relevance of raw input features using the weights assigned to various TimeNet features (refer Equation 2): For example, as shown in Figure 2, we obtain highest relevance scores for Glucose Level (feature 1) and Systolic Blood Pressure (feature 20) for Diabetes Mellitus with Complications (Figure 2(a)), and Essential Hypertension (Figure 2(b)), respectively. Refer Supplementary Material Figure 3 for more details. We conclude that *even though TimeNet was never trained on MIMIC-III data, it still provides meaningful general-purpose features from time series of raw input features, and LASSO helps to select the most relevant ones for end-task by using labeled data.* Further, extracting features using a deep recurrent neural network model for time series of each raw input feature independently – rather than considering a multivariate time series – eventually allows to easily assign relevance scores to raw features in the input domain, allowing a high-level basic model validation by domain-experts.

6 Discussion and Future Work

In this work, we leverage deep learning models efficiently via TimeNet for phenotyping and mortality prediction tasks, with little hyperparameter tuning effort. TimeNet-based features can be efficiently transferred to train linear interpretable classifiers for the end tasks considered while still achieving classification performance similar to more compute-intensive deep models trained from scratch. In future, evaluating a domain-specific TimeNet-like model for clinical time series (e.g. trained only on MIMIC-III database) will be interesting.

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Bengio, 2012] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36, 2012.

[Che *et al.*, 2016] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*, 2016.

[Chen *et al.*, 2015] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, et al. The ucr time series classification archive, July 2015. `www.cs.ucr.edu/~eamonn/time_series_data/`.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[Choi *et al.*, 2016] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318, 2016.

[Harutyunyan *et al.*, 2017] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *arXiv preprint arXiv:1703.07771*, 2017.

[Hermans and Schrauwen, 2013] Michiel Hermans and Benjamin Schrauwen. Training and analysing deep recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 190–198, 2013.

[Johnson *et al.*, 2016] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[Lipton *et al.*, 2015] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[Malhotra *et al.*, 2015] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *ESANN, 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 89–94, 2015.

[Malhotra *et al.*, 2017] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. TimeNet: Pre-trained deep recurrent neural network for time series classification. In *25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2017.

[Micenková *et al.*, 2013] Barbora Micenková, Xuan-Hong Dang, Ira Assent, and Raymond T Ng. Explaining outliers by subspace separability. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 518–527. IEEE, 2013.

[Miotto *et al.*, 2016] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6:26094, 2016.

[Nguyen *et al.*, 2017] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: A convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1):22–30, 2017.

[Purushotham *et al.*, 2017] Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmark of deep learning models on large healthcare mimic datasets. *arXiv preprint arXiv:1710.08531*, 2017.

[Rajkomar *et al.*, 2018] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Peter J Liu, Xiaobing Liu, Mimi Sun, Patrik Sundberg, Hector Yee, et al. Scalable and accurate deep learning for electronic health records. *arXiv preprint arXiv:1801.07860*, 2018.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Song *et al.*, 2017] Huan Song, Deepta Rajan, Jayaraman J Thiagarajan, and Andreas Spanias. Attend and diagnose: Clinical time series analysis using attention models. *arXiv preprint arXiv:1711.03905*, 2017.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[Tibshirani, 1996] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Table 2: Phenotype-wise Classification Performance in terms of AUROC.

S.No.	Phenotype	LSTM-Multi	TimeNet-48	TimeNet-All	TimeNet-48-Eps	TimeNet-All-Eps
1	Acute and unspecified renal failure	0.8035	0.7861	0.7887	0.7912	0.7941
2	Acute cerebrovascular disease	0.9089	0.8989	0.9031	0.8986	0.9033
3	Acute myocardial infarction	0.7695	0.7501	0.7478	0.7533	0.7509
4	Cardiac dysrhythmias	0.684	0.6853	0.7005	0.7096	0.7239
5	Chronic kidney disease	0.7771	0.7764	0.7888	0.7960	0.8061
6	Chronic obstructive pulmonary disease and bronchiectasis	0.6786	0.7096	0.7236	0.7460	0.7605
7	Complications of surgical procedures or medical care	0.7176	0.7061	0.6998	0.7092	0.7029
8	Conduction disorders	0.726	0.7070	0.7111	0.7286	0.7324
9	Congestive heart failure; nonhypertensive	0.7608	0.7464	0.7541	0.7747	0.7805
10	Coronary atherosclerosis and other heart disease	0.7922	0.7764	0.7760	0.8007	0.8016
11	Diabetes mellitus with complications	0.8738	0.8748	0.8800	0.8856	0.8887
12	Diabetes mellitus without complication	0.7897	0.7749	0.7853	0.7904	0.8000
13	Disorders of lipid metabolism	0.7213	0.7055	0.7119	0.7217	0.7280
14	Essential hypertension	0.6779	0.6591	0.6650	0.6757	0.6825
15	Fluid and electrolyte disorders	0.7405	0.7351	0.7301	0.7377	0.7328
16	Gastrointestinal hemorrhage	0.7413	0.7364	0.7309	0.7386	0.7343
17	Hypertension with complications and secondary hypertension	0.76	0.7606	0.7700	0.7792	0.7871
18	Other liver diseases	0.7659	0.7358	0.7332	0.7573	0.7530
19	Other lower respiratory disease	0.688	0.6847	0.6897	0.6896	0.6922
20	Other upper respiratory disease	0.7599	0.7515	0.7565	0.7595	0.7530
21	Pleurisy; pneumothorax; pulmonary collapse	0.7027	0.6900	0.6882	0.6909	0.6997
22	Pneumonia	0.8082	0.7857	0.7916	0.7890	0.7943
23	Respiratory failure; insufficiency; arrest (adult)	0.9015	0.8815	0.8856	0.8834	0.8876
24	Septicemia (except in labor)	0.8426	0.8276	0.8140	0.8296	0.8165
25	Shock	0.876	0.8764	0.8564	0.8763	0.8562

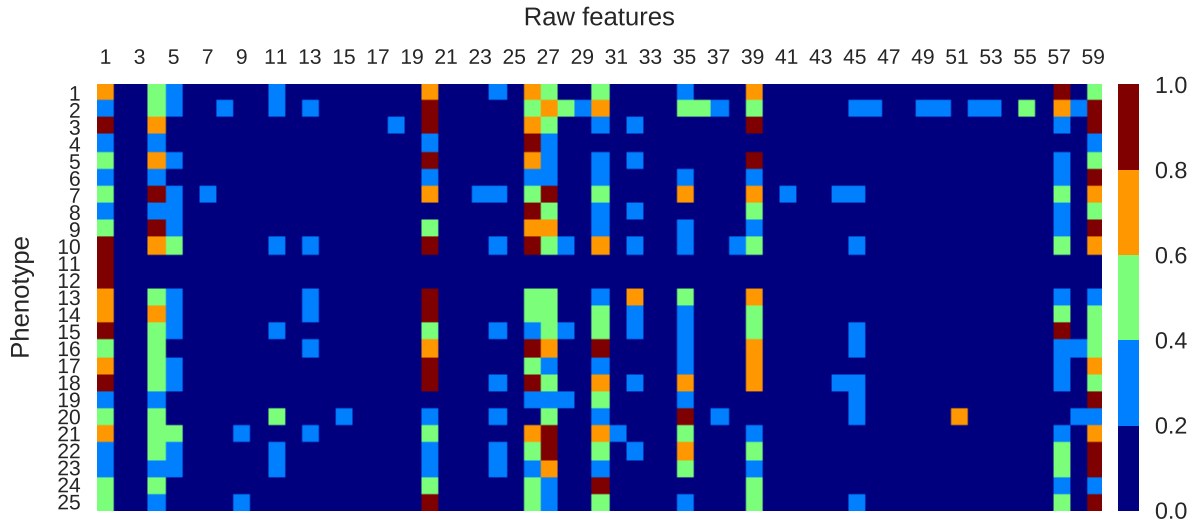


Figure 3: Feature relevance scores for 25 phenotypes. Refer Table 2 for names of phenotypes, and Table 3 for names of raw features.

Table 3: List of raw input features.

1	Glucose	31	Glascow coma scale eye opening → 3 To speech
2	Glascow coma scale total → 7	32	Height
3	Glascow coma scale verbal response → Incomprehensible sounds	33	Glascow coma scale motor response → 5 Localizes Pain
4	Diastolic blood pressure	34	Glascow coma scale total → 14
5	Weight	35	Fraction inspired oxygen
6	Glascow coma scale total → 8	36	Glascow coma scale total → 12
7	Glascow coma scale motor response → Obeys Commands	37	Glascow coma scale verbal response → Confused
8	Glascow coma scale eye opening → None	38	Glascow coma scale motor response → 1 No Response
9	Glascow coma scale eye opening → To Pain	39	Mean blood pressure
10	Glascow coma scale total → 6	40	Glascow coma scale total → 4
11	Glascow coma scale verbal response → 1.0 ET/Trach	41	Glascow coma scale eye opening → To Speech
12	Glascow coma scale total → 5	42	Glascow coma scale total → 15
13	Glascow coma scale verbal response → 5 Oriented	43	Glascow coma scale motor response → 4 Flex-withdraws
14	Glascow coma scale total → 3	44	Glascow coma scale motor response → No response
15	Glascow coma scale verbal response → No Response	45	Glascow coma scale eye opening → Spontaneously
16	Glascow coma scale motor response → 3 Abnorm flexion	46	Glascow coma scale verbal response → 4 Confused
17	Glascow coma scale verbal response → 3 Inapprop words	47	Capillary refill rate → 0.0
18	Capillary refill rate → 1.0	48	Glascow coma scale total → 13
19	Glascow coma scale verbal response → Inappropriate Words	49	Glascow coma scale eye opening → 1 No Response
20	Systolic blood pressure	50	Glascow coma scale motor response → Abnormal extension
21	Glascow coma scale motor response → Flex-withdraws	51	Glascow coma scale total → 11
22	Glascow coma scale total → 10	52	Glascow coma scale verbal response → 2 Incomp sounds
23	Glascow coma scale motor response → Obeys Commands	53	Glascow coma scale total → 9
24	Glascow coma scale verbal response → No Response-ETT	54	Glascow coma scale motor response → Abnormal Flexion
25	Glascow coma scale eye opening → 2 To pain	55	Glascow coma scale verbal response → 1 No Response
26	Heart Rate	56	Glascow coma scale motor response → 2 Abnorm extensn
27	Respiratory rate	57	pH
28	Glascow coma scale verbal response → Oriented	58	Glascow coma scale eye opening → 4 Spontaneously
29	Glascow coma scale motor response → Localizes Pain	59	Oxygen saturation
30	Temperature		