Temporal Phenotyping using Deep Predictive Clustering of Disease Progression

Changhee Lee [1]   Mihaela van der Schaar [2 3 1]

Abstract

Due to the wider availability of modern electronic health records, patient care data is often being stored in the form of time-series. Clustering such time-series data is crucial for patient phenotyping, anticipating patients' prognoses by identifying "similar" patients, and designing treatment guidelines that are tailored to homogeneous patient subgroups. In this paper, we develop a deep learning approach for clustering time-series data, where each cluster comprises patients who share similar future outcomes of interest (e.g., adverse events, the onset of comorbidities). To encourage each cluster to have homogeneous future outcomes, the clustering is carried out by learning discrete representations that best describe the future outcome distribution based on novel loss functions. Experiments on two real-world datasets show that our model achieves superior clustering performance over state-of-the-art benchmarks and identifies meaningful clusters that can be translated into actionable information for clinical decision-making.

1. Introduction

Chronic diseases – such as cystic fibrosis and dementia – are heterogeneous in nature, with widely differing outcomes even in narrow patient subgroups. Disease progression manifests through a broad spectrum of clinical factors, collected as a sequence of measurements in electronic health records, which gives a rise to complex progression patterns among patients (Samal et al., 2011; Yoon et al., 2017). For example, cystic fibrosis evolves slowly, allowing for development of comorbidities and bacterial infections, and creating distinct responses to therapeutic interventions, which in turn makes the survival and quality of life substantially different (Ramos et al., 2017; Lee et al., 2019). Identifying patient subgroups with similar progression patterns can be advantageous for

understanding such heterogeneous diseases. This allows clinicians to anticipate patients' prognoses by comparing to "similar" patients and to design treatment guidelines tailored to homogeneous subgroups (Zhang et al., 2019).

Temporal clustering has been recently used as a data-driven framework to partition patients with time-series observations into subgroups of patients. Recent research has typically focused on either finding fixed-length and low-dimensional representations (Zhang et al., 2019; Rusanov et al., 2016) or on modifying the similarity measure (Giannoula et al., 2018; Luong & Chandola, 2017) both in an attempt to apply the existing clustering algorithms to time-series observations. However, clusters identified from these approaches are purely unsupervised – they do not account for patients' observed outcomes (e.g., adverse events, the onset of comorbidities, etc.) – which leads to heterogeneous clusters if the clinical presentation of the disease differs even for patients with the same outcomes. Thus, a common prognosis in each cluster remains unknown which can mystify the understanding of the underlying disease progression (Boudier et al., 2019; Wami et al., 2013). To overcome this limitation, we focus on *predictive clustering* (Blockeel et al., 2017) to combine predictions on the future outcomes with clustering. More specifically, we aim at finding cluster assignments and centroids by learning discrete representations of time-series that best describe the future outcome distribution. By doing so, patients in the same cluster share similar future outcomes to provide a prognostic value. Figure 1 illustrates a pictorial depiction of the clustering procedure.

In this paper, we propose an actor-critic approach for temporal predictive clustering, which we call AC-TPC.[1] Our model consists of three networks – an *encoder*, a *selector*, and a *predictor* – and a set of centroid candidates. The key insight, here, is that we model temporal predictive clustering as learning discrete representations of the input time-series that best describe the future outcome distribution. More specifically, the encoder maps an input time-series into a continuous latent encoding; the selector assigns a cluster (i.e., maps to a discrete representation) to which the input belongs by taking the latent encoding as an input; and the predictor estimates the future outcome distributions condi-

---

[1]University of California, Los Angeles, USA [2]University of Cambridge, UK [3]Alan Turing Institute, UK. Correspondence to: Changhee Lee <chl8856@ucla.edu>.
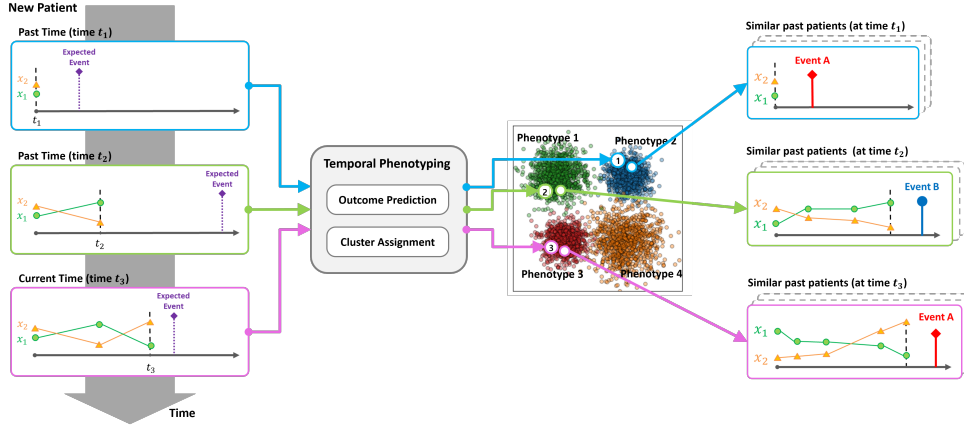
Figure 1. A conceptual illustration of our (real-time) clustering procedure. Here, a new patient is assigned over time to one of the four phenotypes based on the expected future event – either Event A or Event B – as new observations are collected.

tioned on either the encoding or the centroid of the selected cluster (i.e., the selected discrete representation). The following three contributions render our model to achieve our goal. First, to encourage homogeneous future outcomes in each cluster, we define a clustering objective based on the Kullback-Leibler (KL) divergence between the predictor's output given the time-series, and that given the assigned centroids. Second, we transform solving a combinatorial problem of identifying clusters into iteratively solving two sub-problems: optimization of the cluster assignments and optimization of the centroids. Finally, we allow "back-propagation" through the sampling process of the selector by adopting actor-critic training (Konda & Tsitsiklis, 2000).

Throughout the experiments, we show significant performance improvements over the state-of-the-art clustering methods on two real-world medical datasets. To demonstrate the practical significance of our model, we consider a more realistic scenario where the future outcomes of interest are high-dimensional – that is, development of multiple comorbidities in the next year – and interpreting all possible combinations is intractable. Our experiments show that our model can identify meaningful clusters that can be translated into actionable information for clinical decision-making.

2. Problem Formulation

Let $\mathbf{X} \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be random variables for an input feature and an output label (i.e., one or a combination of future outcome(s) of interest) with a joint distribution p_{XY} (and marginal distributions are p_X and p_Y) where $\mathcal{X}$ is the feature space and $\mathcal{Y}$ is the label space. Here, we focus our description on C-class classification tasks, i.e., $\mathcal{Y} = \{1, \cdots, C\}$.[2]

---

[2]In the Supplementary Material, we discuss simple modifications for regression $\mathcal{Y} = \mathbb{R}$ and M-dimensional binary classification tasks $\mathcal{Y} = \{0, 1\}^M$.

We are given a time-series dataset $\mathcal{D} = \{(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}\}_{n=1}^N$ comprising sequences of realizations (i.e., observations) of the pair $(\mathbf{X}, Y)$ for N patients. Here, $(\mathbf{x}_t^n, y_t^n)_{t=1}^{T^n}$ is a sequence of T^n observation pairs that correspond to patient n and $t \in \mathcal{T}^n \triangleq \{1, \cdots, T^n\}$ denotes the time stamp at which the observations are made. From this point forward, we omit the dependency on n when it is clear in the context and denote $\mathbf{x}_{1:t} = (\mathbf{x}_1, \cdots, \mathbf{x}_t)$.

Our aim is to identify a set of K *predictive clusters*, $\mathcal{C} = \{\mathcal{C}(1), \cdots, \mathcal{C}(K)\}$, for time-series data. Each cluster consists of homogeneous data samples, that can be represented by its centroid, based on a certain similarity measure. There are two main distinctions from the conventional notion of clustering. First, we treat subsequences of each times-series as data samples and focus on partitioning $\{\{\mathbf{x}_{1:t}^n\}_{t=1}^{T^n}\}_{n=1}^N$ into $\mathcal{C}$. Hence, we define a cluster as $\mathcal{C}(k) = \{\mathbf{x}_{1:t}^n | t \in \mathcal{T}^n, s_t^n = k\}$ for $k \in \mathcal{K} \triangleq \{1, \cdots, K\}$ where $s_t^n \in \mathcal{K}$ is the cluster assignment for a given $\mathbf{x}_{1:t}^n$. This is to flexibly update the cluster assignment (in real-time) to which a patient belongs as new observations are being accrued over time. Second, we define the similarity measure with respect to the label distribution and associate it with clusters to provide a prognostic value. More specifically, we want the distribution of output label for subsequences in each cluster to be homogeneous and, thus, can be well-represented by the centroid of that cluster.

Let S be a random variable for the cluster assignment – that depends on a given subsequence $\mathbf{x}_{1:t}$ – and $Y|S = k$ be a random variable for the output given cluster k. Then, such property of predictive clustering can be achieved by minimizing the following Kullback-Leibler (KL) divergence: $KL(Y_t|\mathbf{X}_{1:t} = \mathbf{x}_{1:t}\|Y_t|S_t = k)$ for $\mathbf{x}_{1:t} \in \mathcal{C}(k)$ which is defined as $\int_y p(y|\mathbf{x}_{1:t})\big(\log p(y|\mathbf{x}_{1:t}) - \log p(y|s_t)\big)dy$ where $p(y|\mathbf{x}_{1:t})$ and $p(y|s_t)$ are the label distributions conditioned on a subsequence $\mathbf{x}_{1:t}$ and a cluster assignment
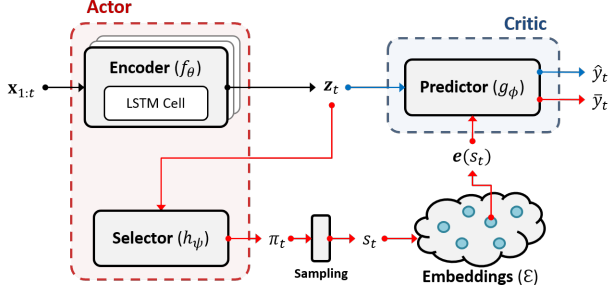
Figure 2. The block diagram of AC-TPC. The red line implies the procedure of estimating $p(y|S_t = s_t)$ via a sampling process and the blue line implies that of estimating $p(y|\mathbf{X}_{1:t} = \mathbf{x}_{1:t})$.

s_t, respectively. Note that the KL divergence achieves its minimum when the two distributions are equivalent.

Finally, we establish our goal as identifying a set of predictive clusters $\mathcal{C}$ that optimizes the following objective:

$$\underset{\mathcal{C}}{\text{minimize}} \sum_{k \in \mathcal{K}} \sum_{\mathbf{x}_{1:t} \in \mathcal{C}(k)} KL\big(Y_t | \mathbf{X}_{1:t} = \mathbf{x}_{1:t} \big\| Y_t | S_t = k\big). \quad (1)$$

Unfortunately, the optimization problem in (1) is highly non-trivial. We need to estimate the objective function in (1) while solving a non-convex combinatorial problem of finding the optimal cluster assignments and cluster centroids.

3. Method: AC-TPC

To effectively estimate the objective function in (1), we introduce three networks – an *encoder*, a *selector*, and a *predictor* – and an *embedding dictionary* as illustrated in Figure 2. These components together provide the cluster assignment and the corresponding centroid based on a given sequence of observations and enable us to estimate the probability density $p(y|s_t)$. More specifically, we define each component as follows:

- The *encoder*, $f_\theta : \prod_{i=1}^{t} \mathcal{X} \to \mathcal{Z}$, is a RNN (parameterized by θ) that maps a (sub)sequence of time-series $\mathbf{x}_{1:t}$ to a latent representation (i.e., encoding) $\mathbf{z}_t \in \mathcal{Z}$ where $\mathcal{Z}$ is the latent space.
- The *selector*, $h_\psi : \mathcal{Z} \to \Delta^{K-1}$, is a fully-connected network (parameterized by ψ) that provides a probabilistic mapping to a categorical distribution from which the cluster assignment $s_t \in \mathcal{K}$ is being sampled.
- The *predictor*, $g_\phi : \mathcal{Z} \to \Delta^{C-1}$, is a fully-connected network (parameterized by ϕ) that estimates the label distribution given the encoding of a time-series or the centroid of a cluster.
- The *embedding dictionary*, $\mathcal{E} = \{\mathbf{e}(1), \cdots, \mathbf{e}(K)\}$ where $\mathbf{e}(k) \in \mathcal{Z}$ for $k \in \mathcal{K}$, is a set of cluster centroids lying in the latent space which represents the corresponding cluster.

Here, $\Delta^{D-1} = \{\mathbf{q} \in [0,1]^D : q_1 + \cdots + q_D = 1\}$ is a $(D-1)$-simplex that denotes the probability distribution for a D-dimensional categorical (class) variable.

At each time stamp t, the *encoder* maps a input (sub)sequence $\mathbf{x}_{1:t}$ into a latent encoding $\mathbf{z}_t \triangleq f_\theta(\mathbf{x}_{1:t})$. Then, based on the encoding $\mathbf{z}_t$, the cluster assignment s_t is drawn from a categorical distribution that is defined by the *selector* output, i.e., $s_t \sim Cat(\pi_t)$ where $\pi_t = [\pi_t(1), \cdots, \pi_t(K)] \triangleq h_\psi(\mathbf{z}_t)$. Once the assignment s_t is chosen, we allocate the latent encoding $\mathbf{z}_t$ to an embedding $\mathbf{e}(s_t)$ in the *embedding dictionary* $\mathcal{E}$. Since the allocated embedding $\mathbf{e}(s_t)$ corresponds to the centroid of the cluster to which $\mathbf{x}_{1:t}$ belongs, we can, finally, estimate the density $p(y|s_t)$ in (1) as the output of the *predictor* given the embedding $\mathbf{e}(s_t)$, i.e., $\bar{y}_t \triangleq g_\phi(\mathbf{e}(s_t))$.

3.1. Loss Functions

In this subsection, we define loss functions to achieve our objective in (1); the details of how we train our model will be discussed in the following subsection.

Predictive Clustering Loss: Since finding the cluster assignment of a given sequence is a probabilistic problem due to the sampling process, the objective function in (1) must be defined as an expectation over the cluster assignment. Thus, we can estimate solving the objective problem in (1) as minimizing the following loss function:

$$\mathcal{L}_1(\theta, \psi, \phi, \mathcal{E}) = \mathbb{E}_{\mathbf{x}, y \sim p_{XY}}\left[\sum_{t \in \mathcal{T}} \mathbb{E}_{s_t \sim Cat(\pi_t)}\big[\ell_1(y_t, \bar{y}_t)\big]\right]$$
(2)

where $\ell_1(y_t, \bar{y}_t) = -\sum_{c=1}^{C} y_t^c \log \bar{y}_t^c$. Here, we slightly abuse the notation and denote $y = [y^1 \cdots y^C]$ as the one-hot encoding of y, and y^c and $\bar{y}^c$ indicates the c-th component of y and $\bar{y}$, respectively. It is worth to highlight that minimizing ℓ_1 is equivalent to minimizing the KL divergence in (1) since the former term of the KL divergence is independent of our optimization procedure.

One critical question that may arise is how to avoid trivial solutions in this unsupervised setting of identifying the cluster assignments and the centroids (Yang et al., 2017). For example, all the embeddings in $\mathcal{E}$ may collapse into a single point or the selector simply assigns equal probability to all the clusters regardless of the input sequence. In both cases, our model will fail to correctly estimate $p(y|s_t)$ and, thus, end up finding a trivial solution. To address this issue, we introduce two auxiliary loss functions that are tailored to address this concern. It is worth to highlight that these loss functions are not subject to the sampling process and their gradients can be simply back-propagated.

Sample-Wise Entropy of Cluster Assignment: To motivate sparse cluster assignment such that the selector ultimately selects one dominant cluster for each sequence, we

introduce sample-wise entropy of cluster assignment which is given as

$$\mathcal{L}_2(\theta, \psi) = \mathbb{E}_{\mathbf{x} \sim p_X} \left[- \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \pi_t(k) \log \pi_t(k) \right] \quad (3)$$

where $\pi_t = [\pi_t(1) \cdots \pi_t(K)] = h_\psi(f_\theta(\mathbf{x}_{1:t}))$. The sample-wise entropy achieves its minimum when π_t becomes an one-hot vector.

Embedding Separation Loss: To prevent the embeddings in $\mathcal{E}$ from collapsing into a single point, we define a loss function that encourages the embeddings to represent different label distributions, i.e., $g_\phi(\mathbf{e}(k))$ for $k \in \mathcal{K}$, from each other:

$$\mathcal{L}_3(\mathcal{E}) = - \sum_{k \neq k'} \ell_1(g_\phi(\mathbf{e}(k)), g_\phi(\mathbf{e}(k'))) \quad (4)$$

where ℓ_1 is reused to quantify the distance between label distributions conditioned on each cluster. We minimize (4) when updating the embedding vectors $\mathbf{e}(1), \cdots, \mathbf{e}(K)$.

3.2. Optimization

The optimization problem in (1) is a non-convex combinatorial problem because it comprises not only minimizing the KL divergence but also finding the optimal cluster assignments and centroids. Hence, we propose an optimization procedure that iteratively solves two subproblems: i) optimizing the three networks – the encoder, selector, and predictor – while fixing the embedding dictionary and ii) optimizing the embedding dictionary while fixing the three networks. Pseudo-code of AC-TPC can be found in the Supplementary Material.

3.2.1. OPTIMIZING THE THREE NETWORK

Finding predictive clusters incorporates the sampling process which is non-differentiable. Thus, to render "back-propagation", we utilize the training of actor-critic models (Konda & Tsitsiklis, 2000). More specifically, we view the combination of the encoder (f_θ) and the selector (h_ψ) as the "actor" parameterized by $\omega_A = [\theta, \psi]$, and the predictor ($g_\phi$) as the "critic". The critic takes as input the the output of the actor (i.e., the cluster assignment) and estimates its value based on the sample-wise predictive clustering loss (i.e., $\ell_1(y_t, \bar{y}_t)$) given the chosen cluster. This, in turn, renders the actor to change the distribution of selecting a cluster to minimize such loss. Thus, it is important for the critic to perform well on the updated output of the actor while it is important for the actor to perform well on the updated loss estimation. As such, the parameters for the actor and the critic need to be updated iteratively.

Given the embedding dictionary $\mathcal{E}$ fixed (thus, we will omit the dependency on $\mathcal{E}$), we train the actor, i.e., the

encoder and the selector, by minimizing a combination of the predictive clustering loss $\mathcal{L}_1$ and the entropy of cluster assignments $\mathcal{L}_2$, which is given by $\mathcal{L}_A(\theta, \psi, \phi) = \mathcal{L}_1(\theta, \psi, \phi) + \alpha \mathcal{L}_2(\theta, \psi)$ where $\alpha \geq 0$ is a coefficient chosen to balance between the two losses. To derive the gradient of this loss with respect $\omega_A = [\theta, \psi]$, we utilize the ideas from actor-critic models (Konda & Tsitsiklis, 2000) in (5) which is displayed at the top; please refer to the Supplementary Material for the detailed derivation. Note that since no sampling process is considered in $\mathcal{L}_2(\theta, \psi)$, we can simply derive $\nabla_{\omega_A} \mathcal{L}_2(\theta, \psi)$.

Iteratively with training the actor, we train the critic, i.e., the predictor, by minimizing the predictive clustering loss $\mathcal{L}_1$ as the following: $\mathcal{L}_C(\phi) = \mathcal{L}_1(\theta, \psi, \phi)$ whose gradient with respect to ϕ can be givens as $\nabla_\phi \mathcal{L}_C(\phi) = \nabla_\phi \mathcal{L}_1(\theta, \psi, \phi)$. Note that since the critic is independent of the sampling process, the gradient can be simply back-propagated.

3.2.2. OPTIMIZING THE CLUSTER CENTROIDS

Now, once the parameters for the three networks (θ, ψ, ϕ) are fixed (thus, we omit the dependency on θ, ψ, and ϕ), we updated the embeddings in $\mathcal{E}$ by minimizing a combination of the predictive clustering loss $\mathcal{L}_1$ and the embedding separation loss $\mathcal{L}_3$, which is given by $\mathcal{L}_E(\mathcal{E}) = \mathcal{L}_1(\mathcal{E}) + \beta \mathcal{L}_3(\mathcal{E})$ where $\beta \geq 0$ is a coefficient chosen to balance between the two losses.

3.2.3. INITIALIZING AC-TPC VIA PRE-TRAINING

Since we transform the combinatorial optimization problem in (1) into iteratively solving two sub-problems, initialization is crucial to achieve better optimization as a similar concern has been addressed in (Yang et al., 2017).

Therefore, we initialize our model based on the following procedure. First, we pre-train the encoder and the predictor by minimizing the following loss function based on the predicted label distribution given the latent encodings of input sequences, i.e., $\hat{y}_t \triangleq g_\phi(\mathbf{z}_t) = g_\phi(f_\theta(\mathbf{x}_{1:t}))$, as the following:

$$\mathcal{L}_I(\theta, \phi) = \mathbb{E}_{\mathbf{x}, y \sim p_{XY}} \left[- \sum_{t \in \mathcal{T}} \ell_1(y_t, \hat{y}_t) \right]. \quad (6)$$

Minimizing (6) encourages the latent encoding to be enriched with information for accurately predicting the label distribution. Then, we perform K-means (other clustering method can be also applied) based on the learned representations to initialize the embeddings $\mathcal{E}$ and the cluster assignments $\{\{s_t^n\}_{t=1}^{T^n}\}_{n=1}^N$. Finally, we pre-train the selector h_ψ by minimizing the cross entropy treating the initialized cluster assignments as the true clusters.

$$\nabla_{\omega_A} \mathcal{L}_A(\theta, \psi, \phi) = \mathbb{E}_{\mathbf{x}, y \sim p_{XY}} \Big[\sum_{t \in \mathcal{T}} \mathbb{E}_{s_t \sim Cat(\pi_t)} \big[\ell_1(y_t, \bar{y}_t) \nabla_{\omega_A} \log \pi_t(s_t) \big] \Big] + \alpha \nabla_{\omega_A} \mathcal{L}_2(\theta, \psi). \tag{5}$$

4. Related Work

Temporal clustering, also known as time-series clustering, is a process of unsupervised partitioning of the time-series data into clusters in such a way that homogeneous time-series are grouped together based on a certain similarity measure. Temporal clustering is challenging because i) the data is often high-dimensional – it consists of sequences not only with high-dimensional features but also with many time points – and ii) defining a proper similarity measure for time-series is not straightforward since it is often highly sensitive to distortions (Ratanamahatana et al., 2005). To address these challenges, there have been various attempts to find a good representation with reduced dimensionality or to define a proper similarity measure for times-series (Aghabozorgi et al., 2015).

Recently, (Baytas et al., 2017) and (Madiraju et al., 2018) proposed temporal clustering methods that utilize low-dimensional representations learned by RNNs. These works are motivated by the success of applying deep neural networks to find "clustering friendly" latent representations for clustering static data (Xie et al., 2017; Yang et al., 2017). In particular, Baytas et al. (2017) utilized a modified LSTM auto-encoder to find the latent representations that are effective to summarize the input time-series and conducted K-means on top of the learned representations as an ad-hoc process. Similarly, (Madiraju et al., 2018) proposed a bidirectional-LSTM auto-encoder that jointly optimizes the reconstruction loss for dimensionality reduction and the clustering objective. However, these methods do not associate a target property with clusters and, thus, provide little prognostic value about the underlying disease progression.

Our work is most closely related to SOM-VAE (Fortuin et al., 2019). This method jointly optimizes a static variational auto-encoder (VAE), that finds latent representations of input features, and a self-organizing map (SOM), that allows to map the latent representations into a more interpretable discrete representations, i.e., the embeddings. However, there are three key differences between our work and SOM-VAE. First, SOM-VAE aims at minimizing the reconstruction loss that is specified as the mean squared error between the original input and the reconstructed input based on the corresponding embedding. Thus, similar to the aforementioned methods, SOM-VAE neither associates future outcomes of interest with clusters. In contrast, we focus on minimizing the KL divergence between the outcome distribution given the original input sequence and that given the corresponding embedding to build association between future outcomes of interest and clusters. Second, to overcome non-differentiability caused by the sampling process

(that is, mapping the latent representation to the embeddings), (Fortuin et al., 2019) applies the gradient copying technique proposed by (van den Oord et al., 2017), while we utilize the training of actor-critic model (Konda & Tsitsiklis, 2000). Finally, while we flexibly model time-series using LSTM, SOM-VAE handles time-series by integrating a Markov model in the latent representations. This can be a strict assumption especially in clinical settings where a patient's medical history is informative for predicting the future clinical outcomes (Ranganath et al., 2016).

5. Experiments

In this section, we provide a set of experiments using two real-world time-series datasets. We iteratively update the three networks – the encoder, selector, and predictor – and the embedding dictionary as described in Section 3.2. For the network architecture, we constructed the encoder utilizing a single-layer LSTM (Hochreiter & Schmidhuber, 1997) with 50 nodes and constructed the selector and predictor utilizing two-layer fully-connected network with 50 nodes in each layer, respectively. The parameters (θ, ψ, ϕ) are initialized by Xavier initialization (Glorot & Bengio, 2010) and optimized via Adam optimizer (Kingma & Ba, 2014) with learning rate of 0.001 and keep probability 0.7. We chose the balancing coefficients $\alpha, \beta \in \{0.001, 0.01, 0.1, 1.0\}$ utilizing grid search that achieves the minimum validation loss in (2); the effect of different loss functions are further investigated in the experiments. Here, all the results are reported using 5 random 64/16/20 train/validation/test splits.

5.1. Real-World Datasets

We conducted experiments to investigate the performance of AC-TPC on two real-world medical datasets; detailed statistics of each dataset can be found in the Supplementary Material.

UK Cystic Fibrosis registry (UKCF)[3]: This dataset records annual follow-ups for 5,171 adult patients (aged 18 years or older) enrolled in the UK CF registry over the period from 2008 and 2015, with a total of 25,012 hospital visits. Each patient is associated with 89 variables (i.e., 11 static and 78 time-varying features), including information on demographics and genetic mutations, bacterial infections, lung function scores, therapeutic managements, and diagnosis on comorbidities. We set the development of different comorbidities in the next year as the label of interest at each time stamp.

---

[3]https://www.cysticfibrosis.org.uk

Alzheimer's Disease Neuroimaging Initiative (ADNI)[4]**:** This dataset consists of 1,346 patients in the Alzheimer's disease study with a total of 11,651 hospital visits, which tracks the disease progression via follow-up observations at 6 months interval. Each patient is associated with 21 variables (i.e., 5 static and 16 time-varying features), including information on demographics, biomarkers on brain functions, and cognitive test results. We set predictions on the three diagnostic groups – normal brain functioning, mild cognitive impairment, and Alzheimer's disease – as the label of interest at each time stamp.

5.2. Benchmarks

We compare AC-TPC with clustering methods ranging from conventional approaches based on K-means to the state-of-the-art approaches based on deep neural networks. All the benchmarks compared in the experiments are tailored to incorporate time-series data as described below:

Dynamic time warping followed by K-means: Dynamic time warping (DTW) is utilized to quantify pairwise distance between two variable-length sequences and, then, K-means is applied (**KM-DTW**).

K-means with deep neural networks: To handle variable-length time-series data, we utilize our encoder and predictor that are trained based on (6) for fixed-length dimensionality reduction. Then, we apply K-means on the latent encodings **z** (**KM-E2P ($\mathcal{Z}$)**) and on the predicted label distributions $\hat{y}$ (**KM-E2P ($\mathcal{Y}$)**), respectively.

Extensions of DCN (Yang et al., 2017): Since the DCN is designed for static data, we replace their static auto-encoder with a sequence-to-sequence network to incorporate time-series data (**DCN-S2S**).[5] To associated with the label distribution, we compare a DCN whose static auto-encoder is replaced with our encoder and predictor (**DCN-E2P**) to focus dimensionality reduction while preserving information for label prediction.

SOM-VAE (Fortuin et al., 2019): We compare with SOM-VAE – though, this method aims at visualizing input – since it naturally clusters time-series data (**SOM-VAE**). In addition, we compare with a variation of SOM-VAE by replacing the decoder with our predictor to find embeddings that capture information for predicting the label (**SOM-VAE-P**). For both cases, we set the dimension of SOM to K.

It is worth highlighting that the label information is provided

for training DCN-E2P, KM-E2P, and SOM-VAE-P while the label information is not provided for training KM-DTW, DCN-S2S, and SOM-VAE. Please refer to the Supplementary Material for the summary of major components of the tested benchmarks and the implementation details.

5.3. Performance Metrics

Clustering Performance: We applied the following three standard metrics for evaluating clustering performances when the ground-truth cluster label is available: *purity score*, *normalized mutual information* (NMI) (Vinh et al., 2010), and *adjusted Rand index* (ARI) (Hubert & Arabie, 1985). More specifically, the purity score assesses how homogeneous each cluster is (ranges from 0 to 1 where 1 being a cluster consists of a single class), the NMI is an information theoretic measure of how much information is shared between the clusters and the labels that is adjusted for the number of clusters (ranges from 0 to 1 where 1 being a perfect clustering), and ARI is a corrected-for-chance version of the Rand index which is a measure of the percentage of correct cluster assignments (ranges from -1 to 1 where 1 being a perfect clustering and 0 being a random clustering).

When the ground-truth label is not available, we utilize the average *Silhouette index* (SI) (Rousseeuw, 1987) which measures how similar a member is to its own cluster (homogeneity within a cluster) compared to other clusters (heterogeneity across clusters). Formally, the SI for a subsequence $\mathbf{x}_{1:t}^n \in \mathcal{C}^k$ can be given as follows: $SI(n) = \frac{b(n)-a(n)}{\max(a(n),b(n))}$ where $a(n) = \frac{1}{|\mathcal{C}^k|-1} \sum_{m \neq n} \|y_t^n - y_t^m\|_1$ and $b(n) = \min_{k' \neq k} \frac{1}{|\mathcal{C}^{k'}|} \sum_{m \in \mathcal{C}^{k'}} \|y_t^n - y_t^m\|_1$. Here, we used the L1-distance between the ground-truth labels of the future outcomes of interest since our goal is to group input subsequences with similar future outcomes.

Prediction Performance: To assess the prediction performance of the identified predictive clusters, we utilized both *area under receiver operator characteristic curve* (AUROC) and *area under precision-recall curve* (AUPRC) based on the label predictions of each cluster and the ground-truth binary labels on the future outcomes of interest. Note that the prediction performance is available only for the benchmarks that incorporate the label information during training.

5.4. Clustering Performance

We start with a simple scenario where the true class (i.e., the ground-truth cluster label) is available and the number of classes is tractable. In particular, we set $C = 2^3 = 8$ based on the binary labels for the development of three common comorbidities of cystic fibrosis – diabetes, ABPA, and intestinal obstruction – in the next year for the UKCF dataet and $C = 3$ based on the mutually exclusive three diagnostic groups for the ADNI dataset. We compare AC-

---

[4] https://adni.loni.usc.edu

[5] This extension is a representative of recent deep learning approaches for clustering of both static data (Xie et al., 2017; Yang et al., 2017) and time-series data (Baytas et al., 2017; Madiraju et al., 2018) since these methods are built upon the same concept – that is, applying deep networks for dimensionality reduction to conduct conventional clustering methods, e.g., K-means.

Table 1. Performance comparison on the UKCF and ADNI datasets.

Dataset	Method	Purity	NMI	ARI	AUROC	AUPRC
UKCF	KM-DTW	0.573±0.01*	0.010±0.01*	0.014±0.01*	N/A	N/A
	KM-E2P ($\mathcal{Z}$)	0.719±0.01*	0.211±0.01*	0.107±0.01*	0.726±0.01*	0.425±0.02*
	KM-E2P ($\mathcal{Y}$)	0.751±0.01*	0.325±0.01*	0.440±0.02*	0.807±0.00*	0.514±0.01*
	DCN-S2S	0.607±0.06*	0.059±0.08*	0.063±0.09*	N/A	N/A
	DCN-E2P	0.751±0.02*	0.275±0.02*	0.184±0.01*	0.772±0.03*	0.487±0.03*
	SOM-VAE	0.573±0.01*	0.006±0.00*	0.006±0.01*	N/A	N/A
	SOM-VAE-P	0.638±0.04*	0.201±0.05*	0.283±0.17†	0.754±0.05*	0.331±0.07*
	Proposed	**0.807±0.01**	**0.463±0.01**	**0.602±0.01**	**0.843±0.01**	**0.605±0.01**
ADNI	KM-DTW	0.566±0.02*	0.019±0.02*	0.006±0.02*	N/A	N/A
	KM-E2P ($\mathcal{Z}$)	0.736±0.03†	0.249±0.02	0.230±0.03†	0.707±0.01*	0.509±0.01
	KM-E2P ($\mathcal{Y}$)	0.776±0.05	0.264±0.07	0.317±0.11	0.756±0.04	0.503±0.04
	DCN-S2S	0.567±0.02*	0.005±0.00*	0.000±0.01*	N/A	N/A
	DCN-E2P	0.749±0.06	0.261±0.05	0.215±0.06†	0.721±0.03†	0.509±0.03
	SOM-VAE	0.566±0.02*	0.040±0.06*	0.011±0.02*	N/A	N/A
	SOM-VAE-P	0.586±0.06*	0.085±0.08*	0.038±0.06*	0.597±0.10†	0.376±0.05*
	Proposed	**0.786±0.03**	**0.285±0.04**	**0.330±0.06**	**0.768±0.02**	**0.515±0.02**

$*$ indicates p-value < 0.01, $†$ indicates p-value < 0.05

TPC against the aforementioned benchmarks with respect to the clustering and prediction performance in Table 1.

As shown in Table 1, AC-TPC achieved performance gain over all the tested benchmarks in terms of both clustering and prediction performance – where most of the improvements were statistically significant with p-value < 0.01 or p-value < 0.05 – for both datasets. Importantly, clustering methods – i.e., KM-DTW, DCN-S2S, and SOM-VAE – that do not associate with the future outcomes of interest identified clusters that provide little prognostic value on the future outcomes (note that the true class is derived from the future outcome of interest). This is clearly shown by the ARI value near 0 which indicates that the identified clusters have no difference with random assignments. Therefore, similar sequences with respect to the latent representations tailored for reconstruction or with respect to the shape-based measurement using DTW can have very different outcomes.

In Figure 3, we further investigate the purity score, NMI, and ARI by varying the number of clusters K from 4 to 16 on the UKCF dataset in the same setting with that stated above (i.e., $C = 8$). Here, the three methods – i.e., KM-DTW, DCN-S2S, and SOM-VAE – are excluded for better visualization. As we can see in Figure 3, our model rarely incur performance loss in both NMI and ARI while the benchmarks (except for SOM-VAE-P) showed significant decrease in the performance as K increased (higher than C). This is because the number of clusters identified by AC-TPC (i.e., the number of activated clusters where we define cluster k is activated if $|\mathcal{C}(k)| > 0$) was the same with C most of the times, while the DCN-based methods identified exactly K clusters (due to the K-means). Since the NMI and ARI are adjusted for the number of clusters, a smaller number of identified clusters yields, if everything being equal, a higher performance. In contrast, while our model achieved the same purity score for $K \geq 8$, the benchmark showed improved performance as K increased since the

purity score does not penalize having many clusters. This is an important property of AC-TPC that we do not need to know a priori what the number of cluster is which is a common practical challenge of applying the conventional clustering methods (e.g., K-means).

The performance gain of our model over SOM-VAE-P (and, our analysis is the same for SOM-VAE) comes from two possible sources: i) SOM-VAE-P mainly focuses on visualizing the input with SOM which makes both the encoder and embeddings less flexible – this is why it performed better with higher K – and ii) the Markov property can be too strict for time-series data especially in clinical settings where a patient's medical history is informative for predicting the future clinical outcomes (Ranganath et al., 2016).

5.5. Multiple Future Outcomes – a Practical Scenario

In this experiment, we focus on a more practical scenario where the future outcome of interest is high-dimensional and, thus, the number of classes based on all the possible combinations of future outcomes becomes intractable. Suppose that we are interested in the development of M comorbidities in the next year whose possible combinations grow exponentially $C = 2^M$. Interpreting such a large number of patient subgroups will be a daunting task which hinders the understanding of underlying disease progression. Since different comorbidities may share common driving factors (Ronan et al., 2017), we hope our model to identify much smaller underlying (latent) clusters that govern the development of comorbidities. Here, to incorporate with M comorbidities (i.e., M binary labels), we redefine the output space as $\mathcal{Y} = \{0, 1\}^M$ and modify the predictor and loss functions, accordingly.

We identified 12 clusters of patients based on the next-year development of 22 different comorbidities in the UKCF dataset and reported 5 clusters in Figure 5 – Cluster 0, 5, 7,
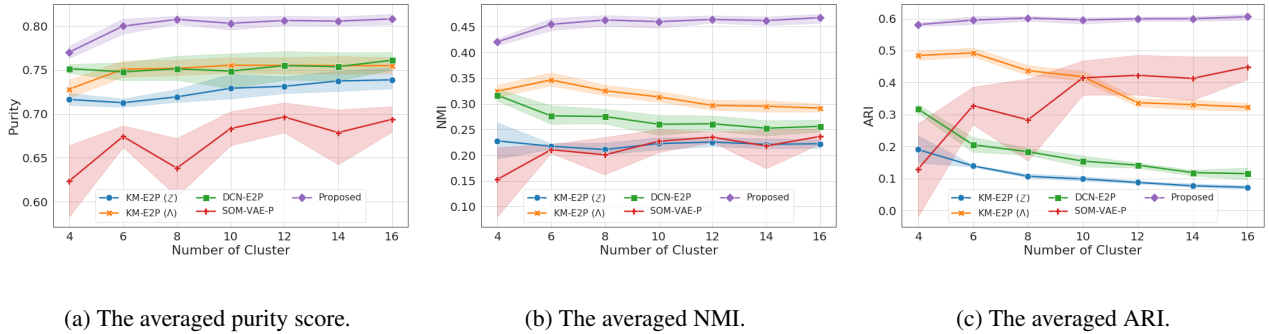
(a) The averaged purity score.

(b) The averaged NMI.

(c) The averaged ARI.

Figure 3. The purity score, NMI, and ARI (mean and 95% confidence interval) for the UKCF dataset ($C = 8$) with various K.
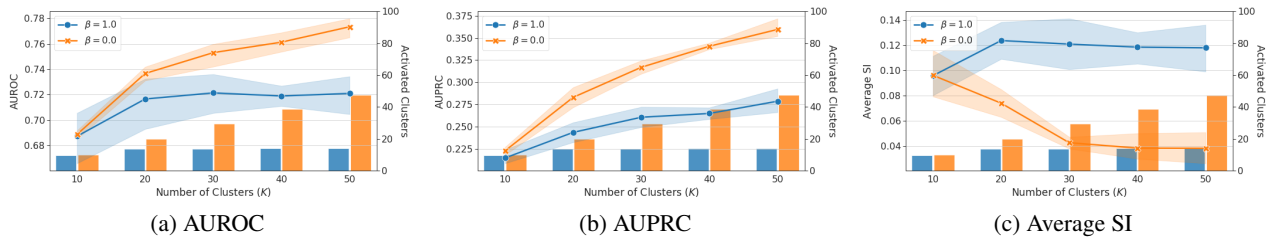

(a) AUROC

(b) AUPRC

(c) Average SI

Figure 4. AUROC, AUPRC, and average SI (mean and 95% confidence interval) and the number of activated clusters with various K.
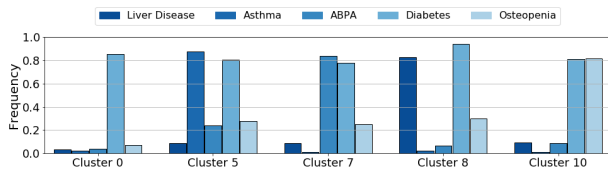

Figure 5. Clusters with high-risk of developing diabetes.

8, and 10 – with the frequency of developing important co-morbidities in the next year. Here, we selected the 5 clusters that have the highest risk of developing diabetes in the next year, and the frequency is calculated in a cluster-specific fashion using the true label. A full list of clusters and co-morbidity frequencies can be found in the Supplementary Material. Although all these clusters displayed high risk of diabetes, the frequency of other co-occurred comorbidities was significantly different across the clusters. In particular, around 89% of the patients in Cluster 5 experienced asthma in the next year while it was less than 3% of the patients in the other cluster. Interestingly, "leukotriene" – a medicine commonly used to manage asthma – and "FEV$_1$% predicted" – a measure of lung function – were the two most different input features between patients in Cluster 5 and those in the other clusters. We observed similar findings in Cluster 7 with ABPA, Cluster 8 with liver disease, and Cluster 10 with osteopenia. Therefore, by grouping patients who are likely to develop a similar set of comorbidities, our method identified clusters that can be translated into

actionable information for clinical decision-making.

5.6. How Does the Temporal Phenotypes Change over Time?

In this subsection, we demonstrate run-time examples of how AC-TPC flexibly updates the cluster assignments over time with respect to the future development of comorbidities in the next year. Figure 6 illustrates three representative patients:

- **Patient A** had diabetes from the beginning of the study and developed asthma as an additional comorbidity at $t = 2$. Accordingly, AC-TPC changed the temporal phenotype assigned to this patient from Cluster 0, which consists of patients who are very likely to develop diabetes but very unlikely to develop asthma in the next year, to Cluster 5, which consists of patients who are likely to develop both diabetes and asthma in the next year, at $t = 1$.

- **Patient B** had ABPA from the beginning of the study and developed diabetes at $t = 5$. Similarly, AC-TPC changed the temporal phenotype assigned to this patient from Cluster 2, which consists of patients who are likely to develop ABPA but not diabetes in the next year, to Cluster 7, which consists of patients who are likely to develop both ABPA and diabetes in the next year, at $t = 4$.
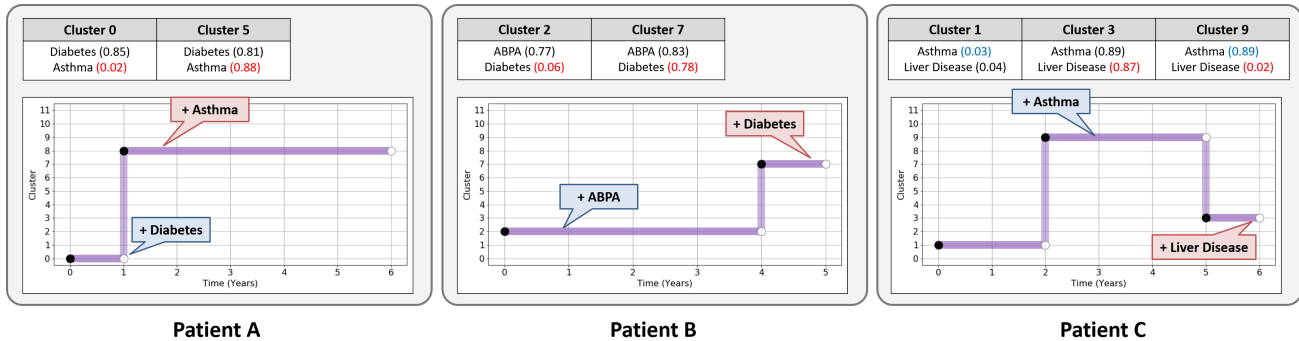
Figure 6. An illustration of run-time examples of AC-TPC on three representative patients.

- **Patient C** had no comorbidity at the beginning of the study, and developed asthma and liver disease as additional comorbidities, respectively at $t = 3$ and $t = 6$. AC-TPC changed the temporal phenotypes assigned to this patient from Cluster 1 to Cluster 9 at $t = 2$ and then to Cluster 3 at $t = 5$. The changes in the temporal phenotypes were consistent with the actual development of asthma and liver disease considering the distribution of comorbidity development in the next year – that is, Cluster 1 consists of patients who are not likely to develop any comorbidities in the next year, Cluster 9 consists of patients who are likely to develop asthma but not liver disease, and Cluster 3 consists of patients who are likely to develop asthma and liver disease in the next year.

5.7. Trade-Off between Clustering and Prediction

In predictive clustering, the trade-off between the clustering performance (for better interpretability) – which quantifies how the data samples are homogeneous within each cluster and heterogeneous across clusters with respect to the future outcomes of interest – and the prediction performance is a common issue. The most important parameter that governs this trade-off is the number of clusters. More specifically, increasing the number of clusters will make the predictive clusters have higher diversity to represent the output distribution and, thus, will increase the prediction performance while decreasing the clustering performance. One extreme example is that there are as many clusters as data samples which will make the identified clusters fully individualized; as a consequence, each cluster will lose interpretability as it no longer groups similar data samples.

To highlight this trade-off, we conduct experiments under the same experimental setup with that of Section 5.5. For the performance measures, we utilized the AUROC and AUPRC to assess the prediction performance, and utilized the average SI to assess the clustering performance. To control the number of activated clusters, we set $\beta = 0$ and $\beta = 1$ (since the embedding separation loss in (4) controls

the activation of clusters) and reported the performance by increasing the number of possible clusters K, i.e., the dimension of the embedding dictionary.

As can be seen in Figure 4, the prediction performance increased with a increasing number of identified clusters due to the higher diversity to represent the label distribution while making the identified clusters less interpretable. That is, the cohesion and separation among clusters become ambiguous as shown in the low average SI. On the other hand, when we set $\beta = 1.0$ (which is selected based on the validation loss in 2), our method consistently identified a similar number of clusters for $K > 20$, i.e., 13.8 on average, in a data-driven fashion and provided slightly reduced prediction performance with significantly better interpretability, i.e., the average SI 0.120 on average.

6. Conclusion

In this paper, we introduced AC-TPC, a deep learning approach for predictive clustering of time-series data. We defined novel loss functions to encourage each cluster to have homogeneous future outcomes (e.g., adverse events, the onset of comorbidities, etc.) and designed optimization procedures to avoid trivial solutions in identifying cluster assignments and the centroids. Throughout the experiments on two real-world datasets, we showed that our model achieves superior clustering performance over state-of-the-art methods and identifies meaningful clusters that can be translated into actionable information for clinical decision-making.

Acknowledgements

References

Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. Time-series clustering – a decade review. *Information Systems*, 53:16–38, May 2015.

Baytas, I. M., Xiao, C., Zhang, X., Wang, F., Jain, A. K., and Zhou, J. Patient subtyping via time-aware lstm networks. *In Proceedings of the 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2017)*, 2017.

Blockeel, H., Dzeroski, S., Struyf, J., and Zenko, B. *Predictive Clustering*. Springer New York, 2017.

Boudier, A., Chanoine, S., Accordini, S., Anto, J. M., na, X. B., Bousquet, J., Demoly, P., Garcia-Aymerich, J., Gormand, F., Heinrich, J., Janson, C., Künzli, N., Matran, R., Pison, C., Raherison, C., Sunyer, J., Varraso, R., Jarvis, D., Leynaert, B., Pin, I., and Siroux, V. Data-driven adult asthma phenotypes based on clinical characteristics are associated with asthma outcomes twenty years later. *Allegy*, 74(5):953–963, May 2019.

Fortuin, V., Hüser, M., Locatello, F., Strathmann, H., and Rätsch, G. SOM-VAE: Interpretable discrete representation learning on time series. *In Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, 2019.

Giannoula, A., Gutierrez-Sacristán, A., Bravo, A., Sanz, F., and Furlong, L. I. Identifying temporal patterns in patient disease trajectories using dynamic ping: A population-based study. *Scientific Reports*, 8(4216):1–14, March 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, 2010.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Hubert, L. and Arabie, P. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. *In Proceedings of the 13th Conference on Neural Information Processing Systems (NIPS 2000)*, 2000.

Lee, C., Yoon, J., and van der Schaar, M. Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Transactions on Biomedical Engineering*, April 2019.

Luong, D. T. A. and Chandola, V. A k-means approach to clustering disease progressions. *In Proceedings of the 5th IEEE International Conference on Healthcare Informatics (ICHI)*, 2017.

Madiraju, N. S., Sadat, S. M., Fisher, D., and Karimabadi, H. Deep temporal clustering: Fully unsupervised learning of time-domain features. *arXiv preprint arXiv:1802.01059*, 2018.

Ramos, K. J., Quon, B. S., Heltshe, S. L., Mayer-Hamblett, N., Lease, E. D., Aitken, M. L., Weiss, N. S., and Goss, C. H. Heterogeneity in survival in adult patients with cystic fibrosis with FEV$_1$ < 30% of predicted in the united states. *Chest*, 151(6):1320–1328, June 2017.

Ranganath, R., Perotte, A., Elhadad, N., and Blei, D. Deep survival analysis. *In Proceedings of the 1st Machine Learning for Healthcare Conference (MLHC 2016)*, 2016.

Ratanamahatana, C. A., Keogh, E., Bagnall, A. J., and Lonardi, S. A novel bit level time series representation with implications for similarity search and clustering. *In Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005)*, 2005.

Ronan, N. J., Elborn, J., and Plant, B. J. Current and emerging comorbidities in cystic fibrosis. *Presse Med.*, 46(6):125–138, June 2017.

Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20:53–65, 1987.

Rusanov, A., Prado, P. V., and Weng, C. Unsupervised time-series clustering over lab data for automatic identification of uncontrolled diabetes. *In Proceedings of the 4th IEEE International Conference on Healthcare Informatics (ICHI)*, 2016.

Samal, L., Wright, A., Wong, B., Linder, J., and Bates, D. Leveraging electronic health records to support chronic disease management: the need for temporal data views. *Informatics in Primary Care*, 19(2):65–74, 2011.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. *In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

Vinh, N. X., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(1):2837–2854, October 2010.

Wami, W. M., Buntinx, F., Bartholomeeusen, S., Goderis, G., Mathieu, C., and Aerts, M. Influence of chronic comorbidity and medication on the efficacy of treatment in patients with diabetes in general practice. *The British Journal of General Practice*, 63(609):267–273, March 2013.

Xie, J., Girshick, R., and Farhadi, A. Unsupervised deep embedding for clustering analysis. *In Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, 2017.

Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. *In Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, 2017.

Yoon, J., Davtyan, C., and van der Schaar, M. Discovery and clinical decision support for personalized healthcare. *IEEE J Biomed Health Inform.*, 21(4):1133–1145, 2017.

Zhang, X., Chou, J., Liang, J., Xiao, C., Zhao, Y., Sarva, H., Henchcliffe, C., and Wang, F. Data-driven subtyping of parkinson's disease using longitudinal clinical records: A cohort study. *Scientific Reports*, 9(797):1–12, January 2019.