A Deep Learning Model for Risk Prediction and Subtyping with Temporal Clinical Data

Regular Paper

Introduction

Risk models play a crucial role in preventing illnesses in hospitals, particularly in intensive care units (ICU), where patients are vulnerable to several complications. When informed of unforeseen potential risks, clinical staff may be able to prevent the disease or be better prepared to care for the patient. As a result, many risk models have been developed for various diseases e.g., sepsis (Henry et al. 2015), pneumonia (Carmo et al. 2021). Prevention of major illnesses can have significant impact on both clinical and economic of healthcare. For example, Paolli et al. (2018) show that just 1% reduction in sepsis can approximately reduce hospital costs in USA by about $450M (by 2016 estimates).

Electronic Medical Records (EMR), containing clinical records of patients, contain valuable information to develop predictive models for various risks. Previous studies, e.g., Ibrahim et al. (2020), have shown that patient populations afflicted by a disease (such as sepsis) show substantial heterogeneity. Distinct subpopulations, called subtypes, having relatively homogeneous clinical characteristics within the larger population can be found for most diseases. Accounting for such subtypes during risk modeling has been shown to improve predictive accuracy (Ibrahim et al. 2020). Moreover, such models also enable personalization of subsequent clinical decision making to each subtype.

Previous approaches to account for subtypes during risk modeling broadly fall into two categories. The first includes a 'cluster-then-predict' approach, where clustering is first done independently to find subtypes and then predictive models (e.g., binary classifiers) for each of these clusters are trained. Since clustering is performed independent of classifier training, such an approach may not discover latent cluster structures in high-dimensional EMR data that are beneficial for subsequent classification. This has led to the development of the second category of approaches that perform simultaneous clustering and classification. State-of-the-art approaches in this category include GRASP (Zhang et al. 2021) and DICE (Huang et al 2021). These approaches, in general, are found to outperform cluster-than-predict approaches. However, they enforce stratification through homogeneous risk outcomes in each cluster. We empirically observe that this additional constraint on the subtype reduces the accuracy of risk prediction. Further, subtypes may be homogeneous with respect to any clinical characteristic, not necessarily the risk and such constraints may not lead to clinically meaningful subtypes.

We address this gap by designing a model that learns the underlying clinical heterogeneity and effectively utilizes it to improve risk prediction. We adopt the design science paradigm (Hevner et al. 2008) to design a new IT artifact, a predictive model based on deep neural networks. Following the guidelines of computational design science (Rai 2017), our artifact's design is motivated by key domain characteristics, i.e., the need for modeling heterogeneity in patient data for clinical risk modeling in and its utility and superiority over previous artifacts – state-of-the-art models for simultaneous risk prediction and patient subtyping – is empirically demonstrated using suitable metrics in the context of two clinically important ICU complications.

Our model, called Multinet, is a subtype-aware risk modeling approach that simultaneously learns the underlying heterogeneity and effectively utilizes it to improve risk prediction. Leveraging the representation learning power of deep neural networks, Multinet finds latent well-clustered representations from temporal clinical data, and, in tandem, cluster-specific classification networks are trained to predict risk outcomes. By training in an end-to-end manner latent representations are learnt based on feedback from the classification networks.

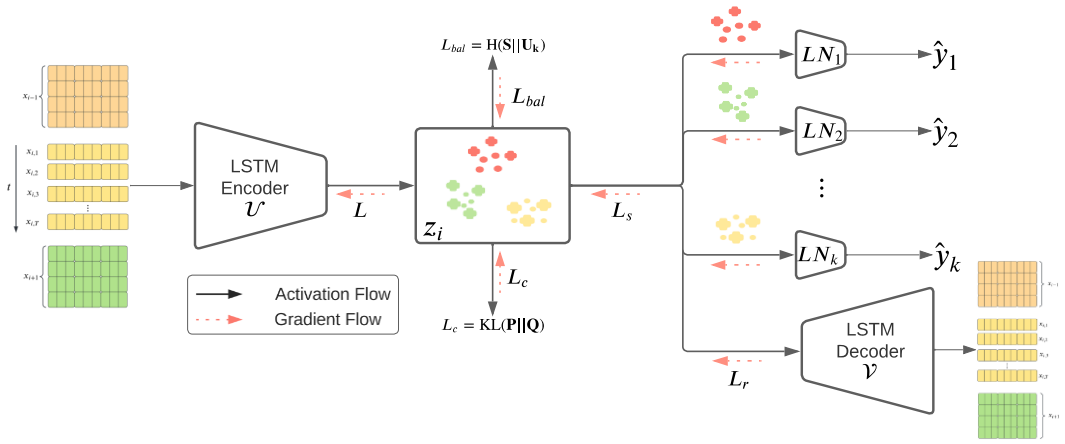
We evaluate Multinet for the task of predicting risk of two life-threatening ICU complications: Sepsis and Acute Respiratory Distress Syndrome (ARDS). Sepsis occurs when the body's response to infection causes tissue damage, organ failure, or death. Globally, in 2017, around 48.9 million developed sepsis, and there were 11 million sepsis-related deaths (Rudd et al. 2020). The costs for managing sepsis in U.S. hospitals – at USD 24 billion annually (13% of U.S. healthcare expenses) – exceed those for any other health condition (Paoli et al. 2018). ARDS often manifests as respiratory failure characterized by rapid onset of widespread inflammation in the lungs. Globally, ARDS affects more than 3 million annually, contributing to about 10% of ICU admissions; with a high mortality of 35-46% (Fan et al. 2018).

Problem Formulation

Electronic Medical Records (EMR) contain clinical data of patients, such as lab measurements and prescribed medications. Most of the data is temporal due to repeated measurements to monitor the health of patients. Given a sequence of such patient records $\boldsymbol{X} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N})$, where $\boldsymbol{x_i}(\boldsymbol{i} = \boldsymbol{1}, \dots, \boldsymbol{N})$ is a multi-dimensional vector representing an inpatient clinical event record (for N patients) at time step $\boldsymbol{t}$. Let $\boldsymbol{x_{i,t}}$ ($\in \mathbb{R}^{\boldsymbol{N_r}}$) represent the $\boldsymbol{t^{th}}$ record of the $\boldsymbol{i^{th}}$ patient. Each EMR record contains $\boldsymbol{N_r}$ features (e.g., lab measurements, indicators for diagnosed diseases and prescribed medications). Given, class labels $\boldsymbol{y_i} \in \{\boldsymbol{1}, \dots, \boldsymbol{B}\}$ associated with every patient record $\boldsymbol{x_i}$ and the number of clusters $\boldsymbol{k}$, our aim is to simultaneously (i) cluster the N time series patient records into $\boldsymbol{k}$ clusters, each represented by a centroid $\boldsymbol{\mu_j} \in \{\boldsymbol{1}, \dots, \boldsymbol{k}\}$, and (ii) build $\boldsymbol{k}$ distinct supervised classification models for each cluster. Note that during training, labels are used only for building the classification models and not for clustering.

Our Approach: Multinet

Fig. 1 shows the neural architecture of Multinet that consists of an LSTM based encoder/decoder and $\boldsymbol{k}$ local networks (LN_j). The encoder $f(U)\colon \mathbb{R}^{\boldsymbol{N_r} \times T} \rightarrow Z$, $Z \in \mathbb{R}^{N_e}$ takes in T patient records and outputs a low-dimensional representation (of size N_e) of the input datapoints. Cluster structure is learned in this latent space and representations in each cluster are used in local networks, $h(W_j)\colon Z \rightarrow \hat{y_j}$ for $j = \{\boldsymbol{1}, \dots, \boldsymbol{k}\}$, to train $\boldsymbol{k}$ classification models. In addition, there is an LSTM



decoder $g(V): Z \rightarrow X$ that is used to reconstruct the input sequence from the embeddings. Multinet is parameterized by network weights: $U, V, \{W_j\}_{j=1}^{k}$, which are learnt by optimizing a combination of losses as described below.

LSTM Unit

We now briefly describe the LSTM used in the encoder and decoder. The standard Recurrent Neural Network (RNN) cell maps an input sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ to an output vector sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$ by using hidden states $\boldsymbol{h} = (h_1, h_2, \ldots, h_T)$ through the following model, over time steps $t = 1$ to $t = T$:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

W denotes the weight matrices and b denotes the bias vectors. H is the hidden layer function and T is the length of input sequence. In Multinet, the last hidden state vector h_T is used as the lower dimensional representation of the patient's 24-hour medical history. h_T for all patients forms the lower dimensional vector space Z.

H is chosen to be the Long-Short-Term-Memory (LSTM) cell that consists of a memory cell to store information. The memory vector c_t can be read, written to, or reset at each time step. Thus, the LSTM update takes the form:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$
$$c_t^l = f \cdot c_{t-1}^l + i \cdot g$$
$$h_t^l = o \cdot tanh(c_t^l)$$

i, o and f are the 3 gates vectors which decide whether the memory is updated, reset to zero or whether it is shown to the hidden vector respectively. The entire LSTM cell is differentiable allowing us to calculate its gradient function, and the three gating functions are helpful in reducing the problem of vanishing gradient that is common in RNN training.

Loss Function

The loss function is designed to simultaneously obtain LSTM representations, cluster them in the representation space and learn classifiers for each cluster. For clustering, we follow the methodology of Deep Embedded Clustering (Xie et al. 2016). The overall loss function L is a weighted combination, with coefficients $\beta, \gamma, \delta > 0$, of the reconstruction loss L_r, clustering loss L_c, cluster balance loss L_{bal} and classification loss L_s:

$$L = L_r + \beta * L_c + \gamma * L_s + \delta * L_{bal} \quad (1)$$

L_c is defined as the KL divergence loss between the two distributions P and Q defined as

$$L_c = KL(P||Q) = \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} log \frac{p_{ij}}{q_{ij}} \quad (2)$$

where q_{ij} is the probability of assigning the i^{th} embedded data point (z_i) to the j^{th} cluster (with centroid μ_j), i.e., the soft cluster assignment values for the i^{th} data point. It is measured using the similarity (via the Student's t kernel) between the embedded point and the centroid (V. D. Maaten and Hinton 2008) $q_{ij} = \frac{\left(1+|z_i-\mu_j|^2\right)^{-1}}{\sum_j\left(1+|z_i-\mu_j|^2\right)^{-1}}$ (3). The target distribution p_{ij} is defined in terms of q_{ij} itself as: $p_{ij} = \frac{q_{ij}^2/\sum_i q_{ij}}{\sum_j q_{ij}^2/\sum_i q_{ij}}$ (4).

The cluster membership distribution Q (Eq. 3) uses representations z_i and cluster centroids μ_j inferred during Multinet training. Following Guo et al. (2017), to improve clustering performance, we add the reconstruction loss measured by mean squared error:

$$L_r = \sum_{i=1}^{N} ||x_i - g(f(x_i))||^2 \quad (5)$$

In such approaches, the encoder can map the centroids to a single point to make the loss zero and thus collapse the clusters resulting in trivial solutions. To address this problem in Multinet we add a cluster balance loss to discourage unevenly distributed cluster sizes. We define the 'soft' size of a cluster C_j as $|C_j| = \sum_i q_{ij}$ and cluster support counts $S = [|C_1|, |C_2|, \dots, |C_k|]$ as a $\boldsymbol{k}$-dimensional probability distribution. Let $\boldsymbol{U_k} = {}^1/_k\,\boldsymbol{U}(0,1)$ denote the $\boldsymbol{k}$-dimensional uniform distribution function. We use the Hellinger distance (H), which measures the dissimilarity between two distributions, as the loss $L_{bal} = H(S||U_k) = \frac{1}{2}||\sqrt{S} - \sqrt{U_k}||$. L_s is a weighted cross entropy loss described in the following section.

Multinet Training

We first pre-train the encoder and decoder with the input data using only the reconstruction loss L_r in order to initialize the parameters $(\boldsymbol{U}, \boldsymbol{V})$, This is followed by $\boldsymbol{k}$-means clustering on $\{\boldsymbol{z_i} = \boldsymbol{f}(\boldsymbol{x_i}; \boldsymbol{U})\}_{\boldsymbol{i=1}}^{\boldsymbol{N}}$ to obtain cluster centroids $\{\boldsymbol{\mu_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{k}}$ which are used to calculate cluster membership and target distributions, $\boldsymbol{Q}, \boldsymbol{P}$. After initialization, we train the complete network using mini-batch Stochastic Gradient Descent and the loss $\boldsymbol{L}$. To stabilize training, we update $\boldsymbol{P}$ only after every epoch.

Since the embeddings are gradually updated throughout each iteration of the main training loop, we train the LNs for more sub-iterations inside each iteration. The number of sub-iterations gradually increases (by 1 for every 5 epochs until a max limit we set to 10). This allows the $\boldsymbol{LNs}$ to learn better from the stabilized clustered embeddings than from the intermediate representations.

Furthermore, in each sub-iteration, we employ a weighted loss, which results in more robust classifiers, as demonstrated by previous studies. Instead of assigning equal weight to all data points when training each LN, we use the probabilistic concept of cluster membership to get individual weights for each training point with respect to each LN. This allows points within each cluster to be given a higher weight than ones outside the cluster. The local loss utilizes these weights to calculate the final loss function. The final classification loss is a weighted cross entropy (CE) loss:

$$L_s = \sum_{j=1}^{k} \sum_{p \in C_j} q_{p,j} CE(y_p, h_j(x_p; V_j)) \quad (6)$$

The LNs are trained in this manner for $T_{iter} - 1$ sub-iterations only, without error backpropagation to the encoder. Individual errors from all k LNs are gathered only at the last (T_{iter}^{th})) iteration and backpropagated to the encoder to change the cluster representations accordingly. After training, the encoder network is frozen, and the local networks are finetuned on the latent embeddings using the cluster-weighted classification loss outlined. The entire training procedure is summarized in Algorithm 1.

Once trained, the encoder and local networks can be used to make predictions. For a test point $x^{\wedge}$, the soft cluster probabilities ($q^{\wedge}$) are calculated from Eq. 3 using the cluster centroids (learnt after training) and embedding of the test point (from the encoder). All the local networks collectively predict the class label using the cluster membership probabilities as: $y_p^{\wedge} = \sum_{j=1}^{k} q_{p,j}\ h_j(x_p)$.

Experimental Evaluation

Data

For our experiments, we use de-identified real patient data from Electronic Medical Records of publicly available ICU databases. For sepsis prediction, we use the dataset from Reyna et al. (2020) of 40,366 patients collected from 2 hospitals in USA: Beth Israel Deaconess Medical Center and Emory University Hospital. For ARDS prediction, we use the MIMIC III dataset (Johnson et al. 2016) comprising 33,798 unique patients. Patients with sepsis and ARDS in the data are identified using standard clinical definitions of these illnesses - Sepsis-3 criteria (Seymour et al. 2016) for sepsis and the Berlin criteria (Ferguson et al. 2012) for ARDS. Based on these definitions, hourly

binary labels indicating the presence or absence of the conditions are defined for individual patients.

Algorithm 1: MultiNet Training

Input: Training Data: $X = (x_1, x_2, \cdots, x_N)$ *s.t.* $x_i = \{x_{i,t}\}_{t=1}^{T}$ $(x_{i,t} \in \mathbb{R}^d)$, labels $y^{N\times 1} \in [\mathcal{B}]^N$, k, LSTM encoder: $f(\cdot;\mathcal{U})$, LSTM decoder: $g(\cdot;\mathcal{V})$ and Local Networks $\{h_j(\cdot;\mathcal{W}_j)\}_{j=1}^{k}$

▷ **Initialization**

Pre-train LSTM encoder and decoder $f(\cdot;\mathcal{U})$ & $g(\cdot;\mathcal{V})$ via back-propagating loss in eq. 5

Compute $\mu(C_j)$ $\forall$ C_j *s.t.* $j \in [k]$

Compute matrices Q and P according to eqs. 3 and 4

▷ **Algorithm**

while *Validation AUC increases* **do**

  for *every mini-batch* $\mathcal{X}_b$ **do**

    $\mathcal{Z}_b \leftarrow f(\mathcal{X}_b;\mathcal{U})$

    Calculate Q_b by eq. 3

    for T_{iter} *sub-iterations* **do**

      Sample $\{s_i \sim q_{i\cdot}\}_{i=1}^{|\mathcal{X}_b|}$

      Calculate $\{\mathcal{J}_m = \{t_i : s_i = j\}_{i=1}^{|\mathcal{X}_b|}\}_{j=1}^{k}$

      Train local classifiers $\{h_j\}_{j=1}^{k}$ on $(\mathcal{J}_j, \mathcal{Y}_j)$

      Update $\{\mathcal{W}_j\}_{j=1}^{k}$

    Backpropagate $L = L_r + \beta \cdot L_c + \gamma \cdot L_s + \delta \cdot L_{bal}$ and update $\mu, \mathcal{U}, \mathcal{V}$ and $\mathcal{W}$

  Update P via eq. (3)

▷ **Fine tune Local Classifiers**

For every cluster C_j, train the classifier h_j on $(\mathcal{X}, \mathcal{Y})$ in mini batches.

Output Trained MULTINET model. Cluster centroids $\mu = \{\mu_j\}_{j=1}^{k}$

Prediction Setting

We use the first 24 hours of data (i.e. set $T = 24$) to predict risk for patients (of each condition, separately) in the remaining ICU stay. Subtypes are obtained by clustering the data. Risk prediction

is formulated as a binary classification problem. All patients whose ICU stay is < 24 hours, and those who develop the condition within 24 hours of their ICU stay, are excluded.

Features

In both the datasets, every patient has one record per hour, each record comprising multiple features. For the sepsis dataset, we use all the variables mentioned in Reyna et al. (2020) along with SOFA scores (Vincent et al. 1996). For the ARDS dataset, we include the variables used in previous studies Yang et al. (2017), Zhang (2018) and Dimitrova et al. (2017). Standard preprocessing steps like feature normalization are undertaken to obtain feature vectors for each patient. Categorical variables are converted to binary using one-hot encoding. We perform forward imputation to handle missing values.

Evaluation Details

For both Sepsis and ARDS, the entire dataset is divided into train-validation-test splits in the ratio 72:13:15 (Test data is 15% of the entire dataset and the validation dataset is 15% of the remaining data). All experiments are conducted on three such random splits, and the average results on the held-out test sets are presented. To assess performance on the risk prediction task, the following binary classification metrics are used: (i) Area under the ROC Curve (AUC) and (ii) Area under the Precision Recall Curve (AUPRC). Clustering performance is measured using the Silhouette Score (SIL). We compare Multinet's risk prediction performance with two state-of-the-art methods, DICE and GRASP. We report the results for k = 2,3,4. In our experiments, we set $\delta = 0.1$, $\beta = 2$ and $\gamma = 10$. These were obtained after evaluating the performance of Multinet on a range of values on the validation sets.

Results

The performance of Multinet, DICE and GRASP is shown in Table 1. The performance of DICE and GRASP is inferior to that of Multinet for all values of *k* tested. In both Sepsis and ARDS, the margin of improvement is greater for AUPRC, which is regarded a superior metric in cases of class imbalance since AUPRC correlates better with positive predictive value and better represents feature discrimination (Ozenne et al. 2015). In contrast to Multinet, the performance of these baselines decreases as the number of clusters increases.

			AUPRC			**AUC**			**SIL**	
Dataset	**k**	**DICE**	**GRASP**	**Multinet**	**DICE**	**GRASP**	**Multinet**	**DICE**	**GRASP**	**Multinet**
ARDS	2	0.096	0.498	**0.784**	0.52	0.493	**0.902**	0.36	0.383	**0.623**
ARDS	3	0.118	0.506	**0.78**	0.588	0.519	**0.908**	0.279	0.291	**0.497**
ARDS	4	0.098	0.5	**0.764**	0.535	0.505	**0.901**	0.098	0.261	**0.385**
Sepsis	2	0.091	0.501	**0.693**	0.542	0.491	**0.808**	0.467	0.542	**0.829**
Sepsis	3	0.088	0.507	**0.742**	0.469	0.528	**0.808**	0.42	0.394	**0.446**
Sepsis	4	0.093	0.495	**0.747**	0.478	0.465	**0.824**	0.309	0.28	**0.311**

Conclusion

Our work contributes to the growing literature on predictive analytics in healthcare information systems. Recent works in healthcare IS include risk models for readmission in patients with congestive heart failure (Bardhan et al. 2015) and with chronic diseases (Ben-Assuli and Padman 2020). Unlike these works, our goal is to effectively utilize heterogeneity, manifested as subtypes, for risk modeling; and we use a different, neural network based approach.

Our main contribution is a new model called Multinet to address the modeling limitations of existing methods for combined risk modeling and subtyping. The deep learning architecture of Multinet is simple, powerful, and extensible. Multinet discovers subpopulations in data at the same time, trains numerous local expert networks on them, and then combines the experts to create the final prediction. Multinet clusters patients directly using input clinical data while also utilizing signals from the predictive performance of local expert models. This enables Multinet to obtain latent patient representations with meaningful cluster structures and yield improved risk prediction. Our empirical results demonstrate the superiority of Multinet over state-of-the-art methods for predicting risk of Sepsis and ARDS in Intensive Care Units.

This work can be extended in many ways. In its current implementation, Multinet can model sequential data and make a single risk prediction for every patient. Future work can explore ways to make predictions at an hourly level. This would require separate mechanisms to handle time evolving clusters in the embedded data space. Multinet can also be extended to handle multiple modalities like text and images which are present in Electronic Medical Records. Methods to provide interpretability to the end user can also be explored.

References

Henry, Katharine E., et al. "A targeted real-time early warning score (TREWScore) for septic shock." *Science translational medicine* 7.299 (2015): 299ra122-299ra122.

Carmo, Thomas A., et al. "Derivation and Validation of a Novel Severity Scoring System for Pneumonia at Intensive Care Unit Admission." *Clinical Infectious Diseases* 72.6 (2021): 942-949.

Paoli, Carly J., et al. "Epidemiology and costs of sepsis in the United States—an analysis based on timing of diagnosis and severity level." *Critical care medicine* 46.12 (2018): 1889.

Ibrahim, Zina M., et al. "On classifying sepsis heterogeneity in the ICU: insight using machine learning." *Journal of the American Medical Informatics Association* 27.3 (2020): 437-443.

Zhang, Chaohe, et al. "GRASP: Generic Framework for Health Status Representation Learning Based on Incorporating Knowledge from Similar Patients." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. No. 1. 2021.

Huang, Yufang, et al. "Deep significance clustering: a novel approach for identifying risk-stratified and predictive patient subgroups." *Journal of the American Medical Informatics Association* 28.12 (2021): 2641-2653.

Hevner, Alan R., et al. "Design science in information systems research." *MIS quarterly* (2004)

Rai, Arun. "Editor's comments: Diversity of design science research." *MIS quarterly* 41.1 (2017)

Rudd, Kristina E., et al. "Global, regional, and national sepsis incidence and mortality, 1990–2017: analysis for the Global Burden of Disease Study." *The Lancet* 395.10219 (2020): 200-211.

Fan, Eddy, Daniel Brodie, and Arthur S. Slutsky. "Acute respiratory distress syndrome: advances in diagnosis and treatment." *Jama* 319.7 (2018): 698-710.

Xie, Junyuan, Ross Girshick, and Ali Farhadi. "Unsupervised deep embedding for clustering analysis." *International conference on machine learning*. PMLR, 2016.

Johnson, Alistair EW, et al. "MIMIC-III, a freely accessible critical care database." *Scientific data* 3.1 (2016): 1-9.

Ferguson, Niall D., et al. "The Berlin definition of ARDS: an expanded rationale, justification, and supplementary material." *Intensive care medicine* 38.10 (2012): 1573-1582.

Seymour, Christopher W., et al. "Assessment of clinical criteria for sepsis: for the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)." *Jama* 315.8 (2016): 762-774.

Ozenne, Brice, Fabien Subtil, and Delphine Maucort-Boulch. "The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases." *Journal of clinical epidemiology* 68.8 (2015): 855-859.

Dimitriadou, Evgenia, Andreas Weingessel, and Kurt Hornik. "Voting-merging: An ensemble method for clustering." *International conference on artificial neural networks*. Springer, Berlin, Heidelberg, 2001.

Ben-Assuli, Ofir, and Rema Padman. "Trajectories of Repeated Readmissions of Chronic Disease Patients: Risk Stratification, Profiling, and Prediction." *MIS Quarterly* 44.1 (2020).

Bardhan, Indranil, et al. "Predictive analytics for readmission of patients with congestive heart failure." *Information Systems Research* 26.1 (2015): 19-39.

Yang, Pengcheng, et al. "A new method for identifying the acute respiratory distress syndrome disease based on noninvasive physiological parameters." PloS one 15.2 (2020): e0226962.

Zhang, Zhongheng. "Identification of three classes of acute respiratory distress syndrome using latent class analysis." PeerJ 6 (2018): e4592.