# Deep Learning from Incomplete Data: Detecting Imminent Risk of Hospital-acquired Pneumonia in ICU Patients

**Travis R. Goodwin, PhD, and Dina Demner-Fushman, MD, PhD**
**Lister Hill National Center for Biomedical Communications**
**National Library of Medicine**
**National Institutes of Health**
**Bethesda, MD, USA**

**Abstract**

*Hospital acquired pneumonia (HAP) is the second most common nosocomial infection in the ICU and costs an estimated $3.1 billion annually. The ability to predict HAP could improve patient outcomes and reduce costs. Traditional pneumonia risk prediction models rely on a small number of hand-chosen signs and symptoms and have been shown to poorly discriminate between low and high risk individuals. Consequently, we wanted to investigate whether modern data-driven techniques applied to respective pneumonia cohorts could provide more robust and discriminative prognostication of pneumonia risk. In this paper we present a deep learning system for predicting imminent pneumonia risk one or more days into the future using clinical observations documented in ICU notes for an at-risk population ($n = 1,467$). We show how the system can be trained without direct supervision or feature engineering from sparse, noisy, and limited data to predict future pneumonia risk with 96% Sensitivity, 72% AUC, and 80% $F_1$-measure, outperforming SVM approaches using the same features by 20% Accuracy (relative; 12% absolute).*

## Introduction

The Centers for Disease Control (CDC) estimates that one in every twenty-five acute care hospitalizations results in a health care-associated infection (HAI).[1] Recent analyses indicates that at least 50% of HAIs are preventable[2]. Not only are HAI used to measure quality of care by the Centers for Medicare and Medicaid Services (CMS), failure to prevent HAIs can result in financial penalties such as those described in the 2010 Patient Protection and Affordable Care Act. Moreover, HAIs are estimated to cost over $9.8 billion USD annually, with hospital acquired pneumonia (HAP) costing an estimated $3.1 billion alone.[3] Hospital acquired pneumonia is the second most common nosocomial infection in the intensive care unit (ICU) and the most common in mechanically ventilated patients.[4] Hospital and community acquired pneumonia require different treatment plans (i.e., antibiotics) than other lower respiratory tract infections. Clearly, the ability to predict pneumonia risk can improve patient outcomes by enabling early interventions, monitoring and improved clinical decision support.

A number of pneumonia risk prediction models have been developed which rely on hand-curated signs and symptoms.[5,6] In an external evaluation of these models, Schierenberg et. al. (2016)[6] found that existing pneumonia models typically distinguish between risk and no risk, but fail to discriminate between high- and low-risk patients. Moreover, the majority of models target community acquired pneumonia, with the few HAP prediction models being focused on predicting the risk of HAP following a specific procedure.[7] Consequently, we were interested in evaluating whether data-driven techniques enabled by recent advances in deep learning would enable us to automatically predict when at-risk individuals progress to high- (i.e., "imminent") risk by considering a richer set of clinical observations than those used in prior pneumonia risk prediction models. Specifically, we were interested in harnessing the clinical narratives documented in ICU notes. Clinical narratives provide a rich but underutilized source of information for clinical decision support, potentially documenting and/or summarizing the main observations about the patient, relevant procedures and important positive and/or negative laboratory results. Unfortunately, processing clinical narratives requires overcoming several barriers,[8] including the prevalence of missing,[9] inconsistent, or underspecified information.[10] Moreover, while typical risk predictors such as vital signs are continually recorded through a patient's stay, clinical notes are produced at irregular intervals and often in bursts (that is, there are often multiple days during a patients ICU stay in which no clinical notes are produced). Consequently, inferring pneumonia-risk from clinical notes requires accounting for (1) incomplete or sparse information, (2) gaps in the patient's time line in which no notes were generated, (3) limited availability of training data, and (4) lack of direct (ground-truth) pneumonia risk labels.

In this paper, we present the Pneumonia Risk predictiOn NeTwOrk (PRONTO) which harnesses modern deep learning techniques to infer and predict when and if an at-risk ICU patient will progress to *imminent* pneumonia risk within

a given time window based on the content of his or her longitudinal ICU notes. Specifically, we identified a cohort of at-risk ICU patients ($n = 1,467$) who developed pneumonia during their stay and used an emergent deep learning technique known as Recurrent Additive Networks[11] (RANs) to jointly predict the progression of pneumonia risk, relevant clinical observations, and temporal interactions based on longitudinal analysis of ICU notes generated for a retrospective pneumonia cohort. We show that not only can PRONTO be successfully trained from a limited, sparse (i.e., incomplete), and noisy data set, but that it substantially outperforms SVM-based alternatives by up to 20% (relative; 12% absolute) increased accuracy.

**Background and Significance**

Classic models of disease-risk prediction have relied on a small set of specific risk factors. For example, the Heckerling Clinical Decision Rule for the Diagnosis of Pneumonia evaluates the risk for Pneumonia based on five binary risk factors[5] indicating whether the patient's (1) temperature is above 37.8 °C or (2) heart rate is above 100 bpm as well as whether or not the patient has (3) crackles/rales, (4) decreased breath sounds, or (5) asthma. By comparison, PRONTO considers every risk factor or clinical observation recorded in each clinical note. Consequently, it can be seen as a generalization of these models, though we note that some routine information like temperature may not always be documented in the narrative.

Previous work for disease-risk prediction using natural language processing has largely focused on a small set of features and on classifying risk at the time that each clinical note was written. For example, Bejan et al. (2013)[12] represented patient's clinical history as a sequence of days (with no gaps) where each day was associated with a clinical note. They designed a support vector machine (SVM) for classifying whether a patient is positive or negative for pneumonia at each day based on the clinical notes produced up-to and on that day. By contrast, PRONTO allows for gaps in a patient's record and can predict whether the patient will be positive or negative for pneumonia at given number of days in the patient's future for which no clinical notes (yet) exist.

Risk prediction at arbitrary times was considered in Goodwin and Harabagiu (2015)[13] in which the authors constructed a multi-layer Hidden Markov Model to predict the presence or absence of seven risk factors associated with heart failure based on the previous clinical note. Unfortunately, that methodology requires pre-computing co-occurrence information for all observations, which does not scale well to large numbers of observations. PRONTO, by contrast, is able to consider a large number of unique observations (28,782 in our experiments), and can predict the risk that a patient will develop pneumonia in the future.

**Materials**

Our experiments relied on MIMIC-III,[14,15] a publicly-available critical care database developed by the Massachusetts Institute of Technology (MIT) Lab for Computation Physiology to support research in intelligent patient monitoring. MIMIC-III contains de-identified health data associated with over 40,000 patients. Although MIMIC provides a wealth of structured information including demographics, charts, laboratory tests, medications, and diagnoses, we exclusively relied on the unstructured textual data (i.e., clinical notes). We obtained an initial retrospective cohort of 281,076 patients in MIMIC-III who were coded with a discharge ICD-9 diagnoses indicating community or hospital acquired pneumonia (ICD-9 486). Within this cohort, we considered only ICU stays in which the patient (1) was not admitted with pneumonia (i.e., pneumonia was not community acquired) and (2) was not diagnosed with pneumonia during the first day of their stay (using the approach for detecting pneumonia offset described in subsection *C* of the *Methods* section). This resulted in 1,494 unique ICU stays* for 1,467 patients, with each stay associated with multiple clinical notes generated over multiple days. We created sub-cohorts for training, development, and testing using an 8:1:1 random split (at the patient level).

**Methods**

To account for the sparse or incomplete nature of ICU notes – both in terms of the observations documented within the note and in terms of the gaps in a patient's stay without any notes – we considered an abstract representation of the patient's ICU stay. Specifically we discretized the the patient's ICU stay into discrete, non-overlapping, discontiguous 24-hour windows in which clinical notes were produced. We refer to information documented in each 24-hour window as a *clinical snapshot*, and to the discontinuous sequence of clinical snapshots during a patient's ICU stay as his or her *clinical chronology* (details are provided in subsection *B*).

*Note: we merged re-admissions within a 30 day window into a single ICU stay.

To overcome the noisy and incomplete information associated with clinical narratives we needed to design a predictive model which could: (1) discriminate between relevant and irrelevant clinical observations extracted from clinical notes, (2) learn latent temporal interactions between relevant clinical observations documented on different dates, (3) infer long-distance causal and inhibitive relations between clinical observations, their interactions, and the course of pneumonia. To this end, we present a deep-learning architecture named PRONTO (Pneumonia Risk predictiOn NeTwOrk).

Figure 1 illustrates our three-step approach for pneumonia risk prediction:

**Step 1**: we automatically extract *clinical chronologies* for each patient in our training cohort;

**Step 2**: we train PRONTO to infer the risk of developing pneumonia based on latent temporal, causal, and inhibitive relations encoded in the clinical chronologies extracted for the retrospective pneumonia cohort described above; and

**Step 3**: we apply PRONTO to predict the risk of pneumonia within an arbitrary temporal window for a given patient based on his or her ICU notes.

Below, we (A) define and describe how we automatically extract clinical chronologies from longitudinal ICU notes, (B) present the architecture of PRONTO, (C) describe how the model can be trained from a retrospective patient cohort without direct supervision, and (D) explain how PRONTO can be used to predict pneumonia risk for new patients.

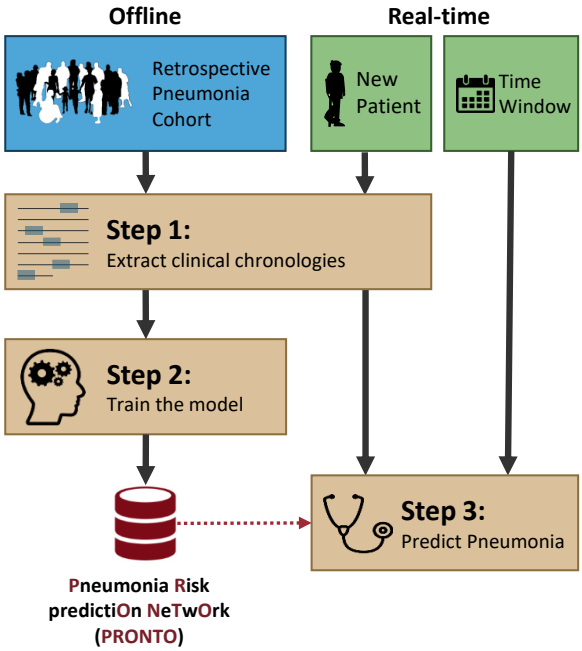



**Figure 1:** Overview of our approach for pneumonia risk prediction

### *A. Extracting Clinical Chronologies*

Extracting the clinical chronology of a patient's ICU stay from clinical notes requires overcoming several barriers. First, both the type of note (e.g., nursing note, radiology, admission report) as well as the frequency of notes varies from day-to-day, admission-to-admission, and patient-to-patient. Second, each note documents a different and incomplete (i.e., sparse) set of observations about the patient's clinical picture at the time the note was written (with different types of notes emphasizing different parts of the patient's clinical picture). Third, there are gaps in the patient's ICU stay in which no notes are generated, and days in which multiple notes are generated.

Consequently we represent a patient's ICU stay as a sparse, discontigous sequence of *clinical snapshots* which are defined as the sets of observations about the patient's clinical picture documented across any ICU notes produced on the same date. Thus, extracting the clinical chronology for a patient's ICU stay reduces to (1) identifying the clinical observations reported in his or her ICU notes, and (2) organizing the observations into a sequence of clinical snapshots.

**Identifying Clinical Observations.** Clinical observations are often documented as multi-word nouns (e.g., "ventilator associated pneumonia"). Consequently, to identify clinical observations we first pre-processed each clinical note using the OpenNLP* sentence splitter, tokenizer, lemmatizer, part-of-speech tagger, and dependency parser. After pre-processing, we identified clinical observations using MetaMap Lite[16] and discarded observations whose semantic types did not correspond to problems, interventions, medications, anatomy, or findings.[17] To detect other attributes, we developed a minor extension to MetaMap Lite based on FastContext,[18] a high-performance re-implementation of ConText.[19] FastContext associated each observation with the following semantic attributes:

- *negation* indicating whether the observation was affirmed or negated in the narrative;
- *certainty* indicating whether the author was certain or uncertain about the observation;
- *temporality* indicating whether the observation occurred in the present, the past, or is hypothetical; and
- *experiencer* indicating whether the the observation was associated with the patient or someone else (e.g., family).

We considered only clinical observations which were affirmed, certain, present, and associated with the patient.

*https://opennlp.apache.org/

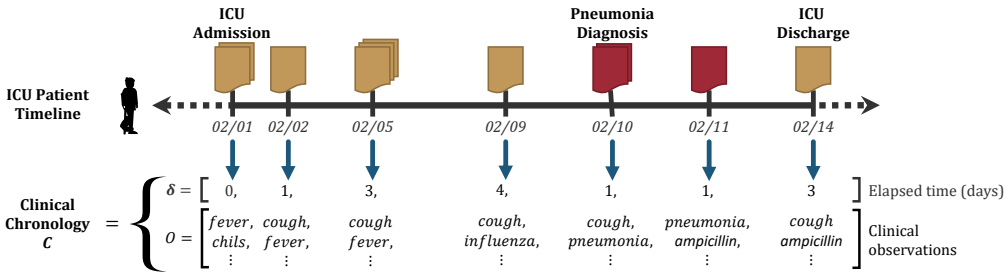



**Figure 2:** Clinical chronology extracted for a single patient's ICU stay with 7 clinical snapshots.

**Mathematical Representation.** We represent the clinical chronology for a patient with $L$ clinical snapshots as a pair $C = (\boldsymbol{\delta}, \boldsymbol{O})$ where $\boldsymbol{\delta} \in \mathbb{Z}^L$ is an elapsed time vector such that $\boldsymbol{\delta}_i$ indicates the elapsed time*between the $i^{\text{th}}$ and $(i-1)^{\text{th}}$ clinical snapshots (where $\boldsymbol{\delta}_0 := 0$), $\boldsymbol{O} \in [1, V]^{L \times N}$ represents the clinical observation matrix such that each column $\boldsymbol{O}_i$ indicates the sequence of clinical observations documented in the $i^{\text{th}}$ clinical snapshot, $V$ is the number of unique clinical observations documented in any clinical note associated with the training cohort, $N$ is the maximum number of observations considered for any clinical snapshot, and $L$ is the maximum number of clinical snapshots used for any clinical chronology. In our experiments, $V = 28,782$, $N := 256$, and $L := 7$.

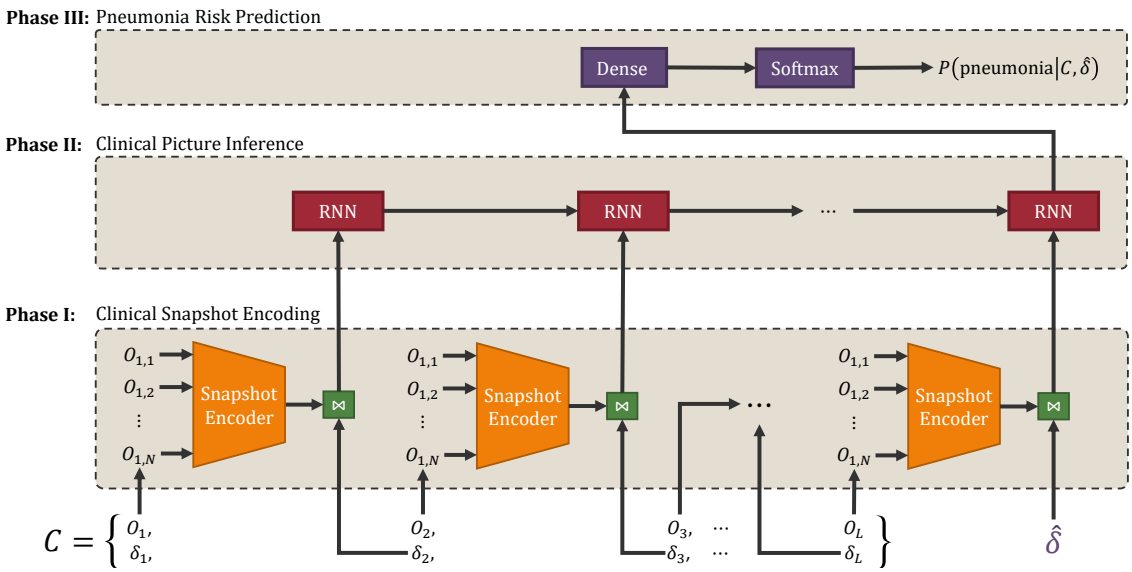



**Figure 3:** Architecture of the Pneumonia Risk predictiOn NeTwOrk (PRONTO), where $C$ indicates the clinical chronology extracted from a patient's ICU stay, $\hat{\delta}$ indicates the prediction window, and $\bowtie$ indicates vector concatenation.

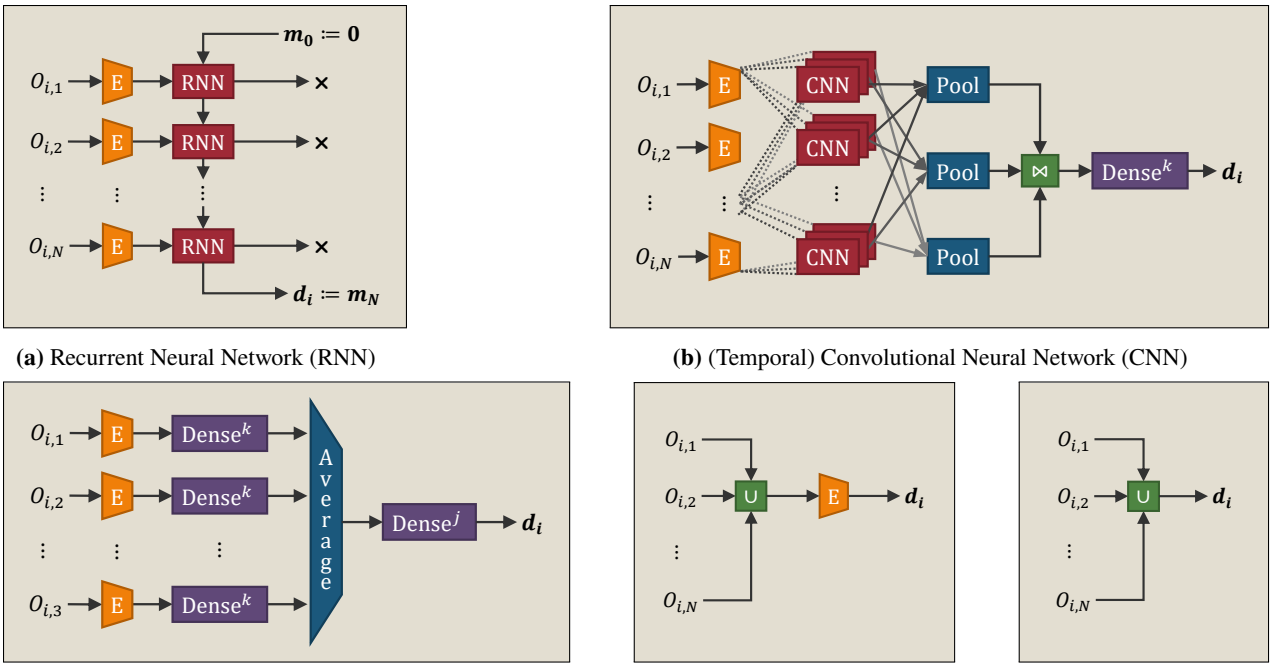
### *B. Architecture of the PRONTO Model*

Given a clinical chronology $C$ and a prediction window $\hat{\delta}$, PRONTO predicts the risk the patient will develop pneumonia during that window using three phases illustrated in Figure 3:

- **Phase I**: **Clinical Snapshot Encoding** processes the sequence of clinical observations $\boldsymbol{O}_i$ in each clinical snapshot $i \in [1, L]$ to produce an embedded representation $\boldsymbol{d}_i$ of the snapshot;
- **Phase II**: **Clinical Picture Inference** processes the encoded clinical snapshots along with their elapsed times to identify latent temporal, causal, and inhibitive interactions and infer the clinical picture of the patient at the end of the given prediction window $\hat{\delta}$; and
- **Phase III**: **Pneumonia Risk Prediction** relies on the inferred clinical picture to predict the probability that the patient will develop pneumonia within $\hat{\delta}$ days.

These three phases are detailed in the remainder of this section.

*To keep observations and the elapsed time within comparable domains (i.e., between 0 and 1), we compute the log of the number of elapsed days and project it between 0 and 1 using the hyperbolic tangent function.




**(a)** Recurrent Neural Network (RNN)

**(b)** (Temporal) Convolutional Neural Network (CNN)

**(c)** Deep Averaging Network (DAN)

**(d)** Dense Encoder

**(e)** Sparse Encoder

**Figure 4:** Neural network architectures for encoding clinical snapshots where $E$ indicates an embedding layer, $\text{Dense}^k$ indicates $k$ consecutive dense layers, $\cup$ indicates binary union, $\bowtie$ indicates vector concatenation, and $\times$ indicates ignored outputs.

**Phase I: Clinical Snapshot Encoding.** The goal of the first phase of PRONTO is to learn an optimal encoding of individual clinical snapshots. Formally, we represent each clinical observation as a $V$-length one-hot vector. We designed PRONTO to use a Deep Averaging Network (DAN) for this purpose. However, due the paucity in the literature for encoding sequences of sparse observations, we implemented and evaluated a total of five neural network architectures for encoding clinical snapshots, as shown in Figure 4. We first present well-known architectures for embedding documents and input sequences: Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs).

*RNN Encoder.* The RNN architecture illustrated in Figure 4a operates by (1) learning an embedded representation of each clinical observation $\boldsymbol{O}_{i,1}; \boldsymbol{O}_{i,2}; \cdots; \boldsymbol{O}_{i,3}$ and (2) sequentially* applying a forward RNN to learn how to compose, combine, or aggregate the embedded clinical observations to produce a single encoding for the entire clinical snapshot by maintaining and updating an internal memory vector. Note: as indicated in Figure 4a, we discard the individual outputs of the RNN and use the final output of the RNN as the clinical snapshot, i.e., $\boldsymbol{d}_i := \boldsymbol{m}_N$. In our experiments, we used Long Short-Term Memory[20] (LSTM) units to build our RNN.

*CNN Encoder.* Like the RNN encoder, the CNN shown in Figure 4b initially learns an embedded representation of each clinical observation. The CNN then combines the embedded representations of adjacent* clinical observations using three parallel convolutional filters, each followed by a maximum pooling layer.† The results of the maximum pooling layers are combined (i.e., concatenated) together and projected into a snapshot embedding using a dense (i.e., fully-connected) Rectified Linear Unit[21] (ReLU). As in Kim (2014),[22] we used three filters operating on sequences of 3, 4, and 5 clinical observations with 1,000 kernels.

*DAN Encoder.* Less well-known than the RNN and CNN, the Deep Averaging Network (DAN) illustrated in Figure 4c also initially learns an embedding for each clinical observation. However, unlike the RNN and CNN, the DAN encoder does not consider the order of clinical observations within a clinical snapshot. The DAN encoder relies on $k$ dense layers to refine the clinical observation embeddings followed by an element-wise average, and $j$ dense layers to refine

*Clinical observations in each clinical snapshot are ordered in the order that they appear in the document. If multiple documents contribute to the same clinical snapshot, the documents are sorted by timestamp, and then by row id (ascending) in MIMIC.

†We also evaluated mean and sum pooling, finding no significant difference between them in our setting.

the average embedding and produce the encoding of the clinical snapshot. As in Goodwin et al. (2017),[23] we used $k := 2$ and $j := 2$ and ReLU activations on all dense layers.

*Dense Encoder.* Like the DAN, the dense projection encoder illustrated in Figure 4d ignores the order of clinical observations in each clinical snapshot. Unlike the DAN, however, the Dense encoder does not learn embeddings for individual clinical observations. Instead, the one-hot representations of each clinical observation are combined using bit-wise union (binary addition) to produce a single, sparse "bag-of-observations" vector. The bag-of-observations is directly embedded to produce the encoding of the clinical snapshot.

*Sparse Encoder.* Finally, the sparse encoder illustrated in Figure 4e does not learn a continuous embedded representation of a clinical chronology. Instead, it uses the un-embedded "bag-of-observations" vector described above as the encoding of the clinical snapshot.

**Phase II: Clinical Picture Inference.** Because the clinical snapshots provide incomplete information about the clinical picture of the patient (e.g., a snapshot produced from a radiology report describing a chest x-ray is unlikely to include many observations about other anatomical regions), the role of Phase II is to infer or impute the clinical picture of the patient *as it relates to imminent pneumonia risk* by combining and accumulating information from each clinical snapshot. We implemented Phase II by (1) casting the inferred clinical picture of the patient as the memory vector of a Recurrent Neural Network (RNN) and (2) training the RNN to infer what the clinical picture of the patient looks like after processing each snapshot and accounting for the elapsed time between it and the previous snapshot. Formally, for each encoded snapshot $\boldsymbol{d}_i \in \boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_L$ the RNN is trained to predict the clinical picture of the patient after elapsed time $\delta_{i+1}$ such that the final output (i.e., inferred clinical picture) computed from $\boldsymbol{d}_L$ and prediction window $\hat{\delta}$ captures sufficient information to predict the risk that the patient will develop pneumonia within $\hat{\delta}$ days.

$$
\begin{aligned}
\widetilde{\boldsymbol{m}}_t &= \boldsymbol{W}_m \boldsymbol{x}_t \\
\boldsymbol{i}_t &= \sigma\left(\boldsymbol{W}_i\left[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\right] + \boldsymbol{b}_i\right) \\
\boldsymbol{f}_t &= \sigma\left(\boldsymbol{W}_f\left[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\right] + \boldsymbol{b}_f\right) \\
\boldsymbol{m}_t &= \boldsymbol{i}_t \circ \widetilde{\boldsymbol{m}}_t + \boldsymbol{f}_t \circ \boldsymbol{m}_{t-1} \\
\boldsymbol{h}_t &= g\left(\boldsymbol{m}_t\right)
\end{aligned}
$$

**Figure 5:** Definition of a Recurrent Additive Network (RAN), where $[\bullet]$ represents vector concatenation, $\widetilde{\boldsymbol{m}}_t$ represents the content layer encoding any new information provided snapshot and elapsed time $\boldsymbol{x}_t$, $\circ$ indicates an element-wise product, $\boldsymbol{i}_t$ represents the input gate, $\boldsymbol{f}_t$ represents the forget gate, $\boldsymbol{m}_t$ represents internal memories about the chronology, $\boldsymbol{h}_t$ is the output layer encoding the accumulated information about clinical picture and therapy as of snapshot $t$ in the patient's chronology, and $g$ is an activation function (in our case, tanh).

The RNN used for inferring the clinical picture is the core of PRONTO; consequently, the choice of memory unit used by the RNN is critical to the over-all success or failure of the model. We used a Recurrent Additive Network[11] (RAN). RANs are a simplified alternative to LSTM- or GRU-based recurrent neural networks that use only additive connections between successive layers and have been shown to obtain similar performance with 38% fewer learnable parameters.[11] The lower number of learnable parameters is ideally suited for deep learning with sparse datasets as it avoids the vanishing gradient problem and lowers the ability of the model to "memorize" the training set, improving generalizability. Formally, let $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ represent the sequential input to the RNN such that $\boldsymbol{x}_t := [\boldsymbol{d}_t, \delta_{t+1}]$ and $\boldsymbol{x}_L := \left[\boldsymbol{d}_L, \hat{\delta}\right]$, let $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_L$ represent the output of the RNN, and let $\boldsymbol{m}_t \in \boldsymbol{m}_1, \cdots, \boldsymbol{m}_L$ represent the internal memory of the RNN after processing $\boldsymbol{x}_t$. Figure 5 provides the equations used to compute $\boldsymbol{h}_t$ with the RAN.

**Phase III: Pneumonia Prediction.** The final phase of PRONTO is to predict the imminent risk (i.e., probability) that the patient will develop pneumonia within prediction window $\hat{\delta}$ using the inferred clinical picture $\boldsymbol{h}_L$. We do this using a dense linear projection layer to produce a two-element vector followed by a softmax activation which computes the probability that the patient is at imminent/high- or low- risk for developing pneumonia within $\hat{\delta}$ days:

$$
P\left(\text{pneumonia} \,\middle|\, C, \hat{\delta}\right) = \text{softmax}\left(\boldsymbol{W}_p \boldsymbol{h}_L + \boldsymbol{b}_p\right) \tag{1}
$$

### *C. Training PRONTO without Direct Supervision*

It is difficult to determine ground-truth labels for risk: risk is inherently difficult for humans to quantify, and using an existing metric of pneumonia risk to train the model would reduce the model to approximating the existing metric rather than learning to distinguish between low- and high-risk. Consequently, in this paper, we used indirect

supervision to train PRONTO. We first detected the onset of pneumonia within the chronology of each patient using the following criteria: (1) if any observation in a *content* section of the note was a descendent of pneumonia in the UMLS hierarchy; (2) if any observation in a *content* section of the note had the word pneumonia or any of its acronyms[24] in its UMLS preferred name; and (3) if there was a non-negated mention of the word pneumonia or any of its acronyms in the *content* section of the note in a line that did not end with a colon. We defined the *content* sections of an ICU note as any section not corresponding to consults, family history, past medical history, or social history. Because some patients have long ICU stays in which the duration of pneumonia is not always clear, we then truncated our chronologies to end at the last snapshot before the onset of pneumonia. In this way, each chronology in the training set begins with one or more snapshots indicating the absence of pneumonia, $\boldsymbol{O}_1, \cdots, \boldsymbol{O}_L$, followed by exactly one snapshot indicating the onset of pneumonia, $\boldsymbol{O}_{L+1}$. These chronologies were used as high-risk examples; i.e., $P\left(\text{pneumonia} \,\middle|\, C = \{\boldsymbol{O}_1, \cdots, \boldsymbol{O}_L; \delta_2, \cdots, \delta_L\}; \hat{\delta} = \delta_{L+1}\right) := 1$ . We created low-risk examples by randomly sampling a sub-sequence length $S \in [1, L-1]$ and truncating each chronology to $S$; i.e., $P\left(\text{pneumonia} \,\middle|\, C = \{\boldsymbol{O}_1, \cdots, \boldsymbol{O}_S; \delta_2, \cdots, \delta_S\}; \hat{\delta} = \delta_{S+1}\right) := 0$ . For example, the chronology illustrated in Figure 2 has the onset of pneumonia in the fifth snapshot and would be associated with two training examples: (1) the chronology including the first four snapshots would be considered as high-risk given the time window $\hat{\delta} = 1$ and (2) a randomly sampled chronology ending at the first, second, or third snapshot would be considered as low-risk, given the time windows $\hat{\delta} \in \{1, 3, 4\}$, respectively. We trained the model by minimizing the cross-entropy loss using Adaptive Moment Estimation[25] (ADAM) with the default initial learning rate $\eta := 0.001$.

*Dropout* The sparse and limited nature of our training data allows the model to avoid learning how to predict pneumonia risk and instead just memorize which chronologies are positive or negative for pneumonia based on rarely occurring observations or sequences of observations (i.e., over-fitting). To prevent over-fitting, we evaluated two forms of dropout: (1) *layer dropout* before each input to the RNN, and between the final output of the RNN and the softmax layer, and (2) *vocabulary dropout* in which random rows (e.g., observations) of the embedding matrix used in the embedding layer of each snapshot encoder was randomly set to a zero vector. Vocabulary-level dropout prevents the model from simply memorizing a rare subset of (confounding) clinical observations that happen to be associated with pneumonia in our dataset.

### *D. Using PRONTO for Inference*

After training PRONTO to predict the disease-risk for a retrospective pneumonia cohort, it can be used to predict the pneumonia-risk for (new) patients in real-time. Specifically, for a (new) patient, we can predict the risk that he or she will develop pneumonia within a given prediction window $\hat{\delta}$ by (1) extracting the clinical chronology from any ICU notes produced thus far in the patients ICU stay, and (2) using Equation (1) to predict the probability of pneumonia given $C$ and $\hat{\delta}$.

## Results

To evaluate PRONTO, we used a test set of 146 at-risk patients as described in the *Materials* section. For each patient, we determined $S$, the snapshot indicating pneumonia onset, and measured the performance of the model when correctly predicting (1) high-risk given the chronology proceeding $S$ and the elapsed time between $S$ and the chronology, and (2) low-risk for a random snapshot occurring in the chronology before $S$. Specific details and examples are provided in subsection C of the *Methods* section. In this way, we produce a test set with an approximately 57% high-risk and 43% low-risk examples and are able to asses that the model can correctly predict when the risk of developing pneumonia progresses from low-risk to high- (imminent) risk.

We compared the performance of PRONTO against a support vector machine (SVM) and two simple baselines:

- **SVM** Inspired by Bejan et al. (2013),[12] we trained a linear SVM to predict pneumonia risk using either (a) only the final snapshot of each training chronology, or (b) using all snapshots in each training chronology (where-in all but the final snapshot are associated with a low risk);
- **Constant** A simple baseline predicting all snapshots as (a) low risk, or (b) high risk; and
- **Random** A simple baseline predicting risk based on (a) the prior probability of high and low risk in the training chronologies, or (b) a uniform distribution over high and low risk

All models were trained for twenty epochs, and the model parameters which resulted in the highest $F_1$-measure on the development set were used for testing.

We measured the Accuracy (A), Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, Precision (P), Recall (R), and $F_1$-measure (the harmonic mean between Precision and Recall).* To investigate the susceptibility of each system to over-fitting, we report all five metrics on the training, development, and testing sets. Table 1 presents the performance of all baseline systems as well as the performance of PRONTO when using each of the clinical snapshot encoding architectures illustrated in Figure 4. To better understand the behavior of the PRONTO model, we also report the learning curve (measured using AUC) on the training, development, and testing data when using each type of clinical snapshot encoder in Figure 6.

**Table 1:** Performance when distinguishing between high/imminent and low pneumonia risk when using (1) Support Vector Machines, (2) constant baselines, (3) random baselines, and (4) PRONTO using different clinical snapshot encoding architectures.

| | **Training** | | | | | **Development** | | | | | **Testing** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **System** | **A** | **AUC** | **P** | **R** | **$F_1$** | **A** | **AUC** | **P** | **R** | **$F_1$** | **A** | **AUC** | **P** | **R** | **$F_1$** |
| SVM: Final Snapshot | 0.97 | 0.96 | 0.96 | **1** | 0.98 | 0.58 | 0.53 | 0.68 | 0.68 | 0.68 | 0.60 | 0.54 | 0.72 | 0.69 | 0.70 |
| SVM: All Snapshots | **1** | **1** | **1** | **1** | **1** | 0.33 | 0.42 | 0.50 | 0.17 | 0.25 | 0.39 | 0.51 | 0.70 | 0.21 | 0.32 |
| Constant: Low Risk | 0.43 | 0.50 | 0 | 0 | 0 | 0.44 | 0.50 | 0 | 0 | 0 | 0.41 | 0.50 | 0 | 0 | 0 |
| Constant: High Risk | 0.57 | 0.50 | 0.57 | **1** | 0.73 | 0.57 | 0.50 | 0.56 | **1** | 0.72 | 0.55 | 0.50 | 0.59 | **1** | 0.74 |
| Random: Prior | 0.54 | 0.49 | 0.66 | 0.65 | 0.66 | 0.55 | 0.51 | 0.67 | 0.64 | 0.66 | 0.54 | 0.47 | 0.67 | 0.66 | 0.66 |
| Random: Uniform | 0.49 | 0.49 | 0.56 | 0.49 | 0.52 | 0.48 | 0.49 | 0.56 | 0.43 | 0.48 | 0.44 | 0.44 | 0.51 | 0.41 | 0.45 |
| PRONTO: RNN | 0.65 | 0.64 | 0.67 | 0.76 | 0.71 | 0.63 | 0.63 | **0.69** | 0.62 | 0.65 | 0.65 | 0.65 | **0.76** | 0.63 | 0.69 |
| PRONTO: CNN | 0.66 | 0.65 | 0.69 | 0.75 | 0.72 | 0.63 | 0.62 | 0.67 | 0.66 | 0.66 | 0.61 | 0.60 | 0.71 | 0.65 | 0.68 |
| ⋆ PRONTO: DAN | 0.67 | 0.64 | 0.65 | **0.87** | 0.75 | **0.71** | **0.68** | 0.66 | **0.96** | **0.78** | **0.72** | **0.66** | 0.69 | **0.96** | **0.80** |
| PRONTO: Dense | **0.81** | **0.80** | **0.81** | 0.87 | **0.84** | 0.56 | 0.55 | 0.56 | 0.74 | 0.64 | 0.59 | 0.55 | 0.63 | 0.74 | 0.68 |
| PRONTO: Sparse | 0.70 | 0.68 | 0.70 | 0.81 | 0.75 | 0.60 | 0.58 | 0.62 | 0.72 | 0.67 | 0.64 | 0.61 | 0.70 | 0.74 | 0.72 |

*Implementation Details* PRONTO was implemented in TensorFlow v1.12.[26] Baseline systems were implemented using Sci-kit Learn[27] v0.19.2. The width of observation embeddings, snapshot embeddings, and the internal memory used by the snapshot RNN were selected from $\{100, 200, 300, 400, 50\}$ and determined to be 200; the number of kernels considered by the CNN was selected from $\{100, 500, 1,000, 2,000\}$ and determined to be $1,000$; the choice of $k$ and $j$ for the CNN and DAN encoders were selected from $\{1, 2, 3\}$ and determined to be 2; the dropout and vocabulary dropout probabilities were selected between 0 and 1 in increments of 0.10 and determined to be 0.0 (no dropout) and 0.50, respectively. Vocabulary size was not limited, the maximum snapshot size was set to 256 (with an average of 141 observations per snapshot), and the maximum chronology length was set to 7 (with an average of 6 snapshots per visit). All baseline and PRONTO experiments used the same fixed random seed.

## Discussion

As shown in Table 1, PRONTO using the Deep Averaging Network (DAN) for encoding clinical snapshots obtained the highest performance, exhibiting a 22% (relative) increase in AUC and a 12% (relative) increase in $F_1$ performance on the testing cohort compared to the best performing SVM (SVM: Final Snapshot). The high performance of PRONTO, especially in terms of Recall, demonstrates that the observations reported in clinical notes alone can be predictive of imminent pneumonia risk. However, the poorer performance of the SVM baselines when using the same observations suggests that deep learning methods can more effectively harness clinical texts even with small datasets. Our results also indicate that model design – particularly in small data settings – has a substantial impact on performance, with sub-optimal models performing similarly to an SVM (in terms of $F_1$). When comparing SVM: All Snapshots to SVM: Final Snapshot, we can see the importance of balancing the distribution of high- and low-risk examples when training the model, indicating that even when the dataset is limited, omitting potential training data can lead to improved generalizability. Interestingly, the dense representation of clinical snapshots provided the highest performance on the training data, but performed poorly compared to other methods on the development and testing set. The DAN, Dense, and Sparse methods viewed observations as unordered set, indicating that the more complex sequential representations may be too expressive to generalize from sparse data.

When examining the learning curves in Figure 6, we can see that all approaches but the DAN quickly over-fit the training

*Note: Due to the class imbalance, we emphasize the $F_1$-measure in our study; the other metrics are reported to illustrate that, unlike the Constant: High Risk baseline, the learned approaches are able to also detect low risk, as evidence by Accuracy and AUC.

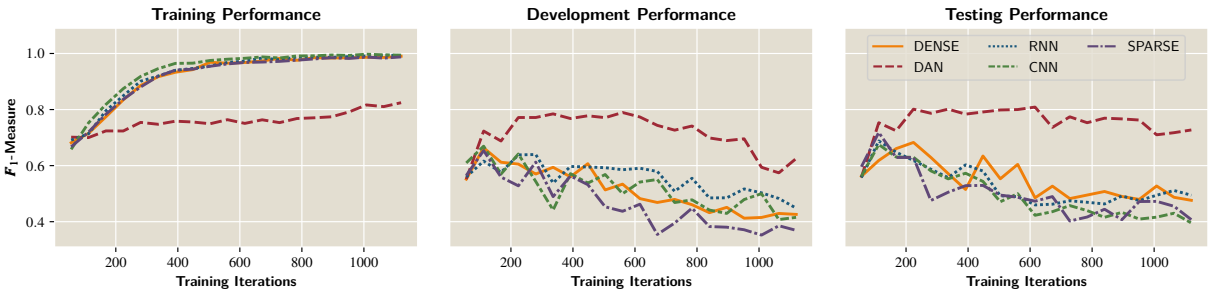



**Figure 6:** $F_1$ learning curve of PRONTO using each clinical snapshot encoder.

data, suggesting that DANs are better able to generalize when used on sparse and limited datasets. Interestingly, despite being the most complex snapshot encoder we considered, the RNN had the lowest performance, suggesting that considering the sequential order of clinical observations within a clinical snapshot had little impact. We believe this is due to the fact that by reducing clinical notes to observations we are removing much of the context around each observation and likely making their order insignificant.

Finally, we examined the accuracy of the disease onset labels we automatically produced and used for indirect supervision. Specifically, we selected difficult-to-label notes (i.e., notes without pneumonia which indicated infiltrates and/or fever and notes with pneumonia without any mention of infiltrates and/or fever) and randomly selected 50 such notes for manual judgment by two physicians. The two physicians had moderate agreement (82% simple agreement; Cohen's $\kappa = 0.49$). We first measured the accuracy of disease onset labels by considering a label as correct if it agreed with either physician, with an accuracy of 80%. We also determined an accuracy of 78.6% when measuring the accuracy of labels only for notes with agreement between both physicians. These results suggest that automatic detection of pneumonia onset with natural language processing is fairly reliable, and well suited for indirect supervision.

### *A. Limitations and Error Analysis*

There are a number of limitations to this study beyond the limited dataset. First, we used automatic natural language processing to identify clinical observations from ICU notes. While UMLS allows some level of normalization, we observed cases in which the clinical observations were underspecified. For example, we found observations such as "aim", "probe", and "distillation" which are ambiguous without context. Second, we found that the attributes detected by ConText were not always reliable, introducing noise into the clinical observations and, indirectly, into the labels used to train the model. In future work we plan to qualify observations by their contextual attributes (experiencer, negation, etc.) to provide more nuanced information to the model. Third, our model considers each ICU stay as an independent chronology. It is possible that information from a previous admission could impact the risk of a patient developing pneumonia (i.e., by allowing the model to distinguish between new observations and re-occurring observations). Fourth, we used an approximately equal number of negative and positive examples when evaluating the model. While this was an intentional decision to make it easier to compare the relative strengths and weaknesses between different models, it does not give an accurate indication of clinical performance where-in one would expect substantially more low-risk patients. Finally, in this initial study we examined only pneumonia. In future work, we would like to predict other types of health-care associated infections, evaluate in a more clinically-realistic setting, and explore the inclusion of charted information.

## Conclusion

In this paper, we described a data-driven deep learning model for distinguishing between low- and high-risk of HAP in an at-risk ICU patient population. We show that not only can our model be successfully trained on incomplete, limited, and noisy data – obtaining 96% Recall, 72% AUC, and 80% $F_1$-measure with less than $1,500$ examples – but that it outperforms SVM-based models using the same features by 22% (AUC). Moreover, we empirically evaluated the impact of five different strategies for encoding clinical observations, determining that Deep Average Networks provide the most reliable encoding of multiple clinical observations. We believe our results demonstrate that ICU notes provide sufficient information to predict and distinguish between high and low pneumonia risk within an arbitrary, given time window without feature engineering or structured data. It is our hope that our findings enable deep learning to be more easily applied in other scenarios with incomplete or limited training data.

## Reproducibility

The source code for PRONTO is available on GitHub at `https://github.com/h4ste/pronto`.


## Acknowledgements

This work was supported by the intramural research program at the U.S. National Library of Medicine, National Institutes of Health.


### References


1. Magill SS, Edwards JR, Bamberg W, Beldavs ZG, Dumyati G, Kainer MA, et al. Multistate Point-Prevalence Survey of Health Care–Associated Infections. N Engl J Med. 2014 Mar;370(13):1198–1208.
2. Schmier JK, Hulme-Lowe CK, Semenova S, Klenk JA, DeLeo PC, Sedlak R, et al. Estimated Hospital Costs Associated with Preventable Health Care-Associated Infections If Health Care Antiseptic Products Were Unavailable. Clinicoecon Outcomes Res. 2016 May;8:197–205.
3. Zimlichman E, Henderson D, Tamir O, Franz C, Song P, Yamin CK, et al. Health Care–Associated Infections: A Meta-Analysis of Costs and Financial Impact on the US Health Care System. JAMA Intern Med. 2013 Dec;173(22):2039–2046.
4. Kalanuria AA, Zai W, Mirski M. Ventilator-Associated Pneumonia in the ICU. Critical Care. 2014 Mar;18(2):208.
5. Heckerling PS, Tape TG, Wigton RS, Hissong KK, Leikin JB, Ornato JP, et al. Clinical Prediction Rule for Pulmonary Infiltrates. Ann Intern Med. 1990 Nov;113(9):664–670.
6. Schierenberg A, Minnaard MC, Hopstaken RM, van de Pol AC, Broekhuizen BDL, de Wit NJ, et al. External Validation of Prediction Models for Pneumonia in Primary Care Patients with Lower Respiratory Tract Infection: An Individual Patient Data Meta-Analysis. PLoS One. 2016 Feb;11(2).
7. Collard HR, Saint S, Matthay MA. Prevention of ventilator-associated pneumonia: an evidence-based systematic review. Annals of Internal Medicine. 2003;138(6):494–501.
8. Weiner M. Evidence Generation Using Data-Centric, Prospective, Outcomes Research Methodologies. San Francisco, CA; 2011.
9. Smith PC, Araya-Guerra R, Bublitz C, Parnes B, Dickinson LM, Van Vorst R, et al. Missing Clinical Information during Primary Care Visits. Jama. 2005;293(5):565–571.
10. Berlin JA, Stang PE. Clinical Data Sets That Need to Be Mined. In: Olsen L, Grossmann C, McGinnis JM, editors. Learning What Works: Infrastructure Required for Comparative Effectiveness Research. vol. 1. Institute of Medicine; 2011. p. 104–114.
11. Lee K, Levy O, Zettlemoyer L. Recurrent Additive Networks; 2017.
12. Bejan CA, Vanderwende L, Evans HL, Wurfel MM, Yetisgen-Yildiz M. On-Time Clinical Phenotype Prediction Based on Narrative Reports. In: AMIA Annual Symposium Proceedings. vol. 2013. American Medical Informatics Association; 2013. p. 103.
13. Goodwin T, Harabiu SM. A Predictive Chronological Model of Multiple Clinical Observations. In: Healthcare Informatics (ICHI), 2015 International Conference On. IEEE; 2015. p. 253–262.
14. Johnson AE, Pollard TJ, Shen L, Li-wei HL, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Scientific data. 2016;3:160035.
15. Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation. 2000;101(23):e215–e220.
16. Demner-Fushman D, Rogers WJ, Aronson AR. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. Journal of the American Medical Informatics Association. 2017;24(4):841–844.
17. Demner-Fushman D, Seckman C, Fisher C, Hauser SE, Clayton J, Thoma GR. A prototype system to support evidence-based practice. In: AMIA Annual Symposium Proceedings. vol. 2008. American Medical Informatics Association; 2008. p. 151.
18. Shi J, Hurdle JF. Trie-based rule processing for clinical NLP: A use-case study of n-trie, making the ConText algorithm more efficient and scalable. Journal of biomedical informatics. 2018;85:106–113.
19. Chapman WW, Hilert D, Velupillai S, Kvist M, Skeppstedt M, Chapman BE, et al. Extending the NegEx lexicon for multiple languages. Studies in health technology and informatics. 2013;192:677.
20. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural computation. 1997;9(8):1735–1780.
21. Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10); 2010. p. 807–814.
22. Kim Y. Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics; 2014. p. 1746–1751.
23. Goodwin TR, Harabagiu SM. Deep Learning from EEG Reports for Inferring Underspecified Information. AMIA Jt Summits Transl Sci Proc. 2017 Jul;2017:112–121.
24. Anand N, Kollef MH. The alphabet soup of pneumonia: CAP, HAP, HCAP, NHAP, and VAP. In: Seminars in respiratory and critical care medicine. vol. 30. Thieme Medical Publishers; 2009. p. 003–009.
25. Kingma D, Ba J. Adam: A Method for Stochastic Optimization. In: Proceedings of the 3rd International Conference on Learning Representations. San Diego; 2015. p. 1–15.
26. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al.. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems; 2015.
27. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-Learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12(Oct):2825–2830.