Visualization of deep models on nursing notes and physiological data for predicting health outcomes through temporal sliding windows

Jienan Yao[1], Yuyang Liu[1], Brenna Li[1], Stephen Gou[1], Chloe Pou-Prom[2], Joshua Murray[2], Amol Verma[2], Muhammad Mamdani[2], and Marzyeh Ghassemi[1]

[1] Department of Computer Science, University of Toronto
{jnyao,yuyang,brli,gouzhen1}@cs.toronto.edu
[2] Unity Health Toronto

Abstract. When it comes to assessing General Internal Medicine (GIM) patients' state, physicians often rely on structured, time series physiological data because it's more efficient and requires less effort to review than unstructured nursing notes. However, these text-based notes can have important information in predicting a patient's outcome. Therefore, in this paper we train two convolutional neural networks (CNN) on in-house hospital nursing notes and physiological data with temporally segmented sliding windows to understand the differences. And we visualize the process in which deep models generate the outcome prediction through interpretable gradient-based visualization techniques. We find that the notes model provides overall better predictions results and it is capable of sending warnings for crashing patients in a more timely manner. Also, to illustrate the different focal points of the models, we identified the top contributing factors each deep model utilizes to make predictions.

Keywords: CNN · EHR · Clinical Notes · Early Warning System · Visualization.

1 Introduction

With recent investments in clinical digitization, many health organizations are interested in developing Clinical Early Warnings Systems (CEWSs) with physiological data and clinicians' notes data, to assist in prioritizing treatment and patient care [4, 18]. However, little is known how these two data types compare relative to clinical predictions. The problem is further complicated because physiological data and textual note data are collected differently and have different temporal segmentation and sparsity [3]. Therefore, we employ the use of sliding windows on both datasets to normalize variability in sparsity and segmentation for more effective comparison. The purpose of this work is to investigate if unstructured textual data can provide comparably valuable information to the structured physiological data (heart rates, oxygen level, lab test results, etc.)

in the context of predicting whether a patient in the General Internal Medicine (GIM) ward would crash[3]. In order to evaluate this, we train machine learning models using either physiological data ("**tabular**" data) or clinical nursing notes ("**notes**" data). We implement convolutional neural net (CNN) models which take as input either of the feature sets (e.g., *tabular* or *notes*), as previous research has shown the effectiveness of CNN architectures in healthcare problems [9, 19, 11]. We then design the following research questions to help us understand how nursing notes and tabular data can be used to predict patient health outcome in GIM: 1) How do our models trained on tabular data compare to models trained on nurse notes data. 2) How early can our deep models make the correct prediction. 3) What information are the deep models using to make the predictions.) 4) What are the top contributing factors of our deep models.

2 Related Work

2.1 Clinical Early Warning Systems

Traditionally, CEWSs take in a set of physiological parameters such as, blood glucose level, oxygen saturation, temperature, etc., with corresponding ranges to calculate a score to determine the patient's health outcomes [18, 16]. However, these traditional CEWSs are dependant on the parameters selected and the method of calculation used, which limits their robustness and predictive power for individual patient encounters [2]. Thus lately, the trend has been to use neural network models in CEWS to account for personalized patient encounter predictions.

2.2 Neural Network Approaches

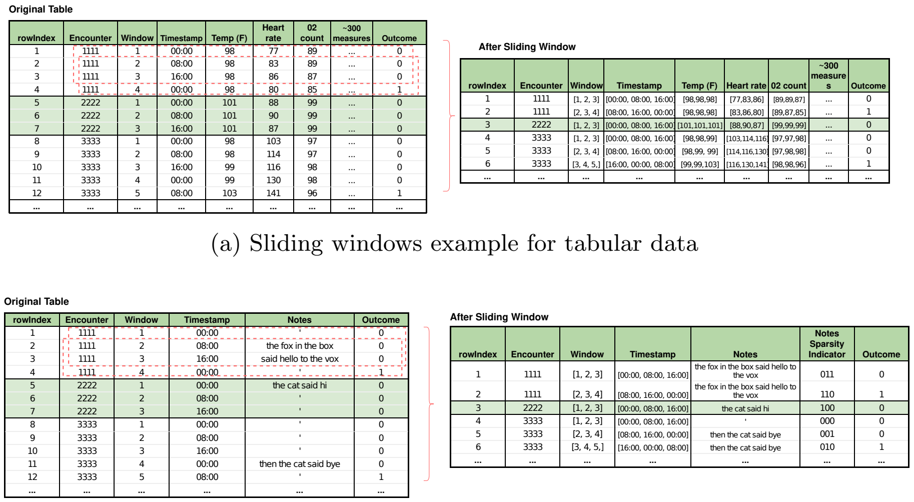
With increasingly publicly available large-scaled health records, such as, the MIMIC II/III dataset [6], opportunities and interests to apply deep learning neural methods to health outcome predictions are rising [4]. Various studies have shown promising results using raw physiological data in predicting patient mortality, ICU transfers, and health deterioration trends [2, 18, 3]. And recently, through advancements in natural language processing, we are also seeing a greater focus on textual data, such as, clinical notes and discharge summaries [8, 5]. As an example, Waudby *et al.* demonstrated that sentiments inferences extracted from the MIMIC III clinical notes can be used as predictors for patient mortality [17]. And other works comparing neural models have found that convolutional neural networks are more suitable for long, unstructured notes [7]. And from these findings, we built our own text extraction CNN model that is fast and easy to tune.

[3] We define a "crashing" patient as someone who is experiencing one of the following outcomes: death on the GIM ward, transfer to intensive care unit (ICU), or transfer to palliative unit.

2.3 Interpreting Prediction Outcomes

A caveat with using deep neural models is the opaqueness in understanding how the machine generates these predictions, especially for clinical notes. It's often unclear which sentences, or words were used and the weight they were given [15]. To visualize the computation behind CNN models on clinical data, [10] applied attention-based deconstruction to emphasize the words contributing to the prediction. However, nursing notes as we know can be very long and heterogeneous, therefore a larger scope on text is needed to make sense behind the reasoning of why it was selected. Therefore, we adapted Grad-CAM, a CNN based visualization technique traditionally on imaging data to our clinical notes [15]. And when applied, with our sliding window approach, as defined in the Method section, allows us to visualize on a time scale the changes in sentences that contribute to the prediction.

3 Methods



Original Table

rowIndex	Encounter	Window	Timestamp	Temp (F)	Heart rate	02 count	~300 measures	Outcome
1	1111	1	00:00	98	77	89	...	0
2	1111	2	08:00	98	83	89	...	0
3	1111	3	16:00	98	86	87	...	0
4	1111	4	00:00	98	80	85	...	1
5	2222	1	00:00	101	88	99	...	0
6	2222	2	08:00	101	90	99	...	0
7	2222	3	16:00	101	87	99	...	0
8	3333	1	00:00	98	103	97	...	0
9	3333	2	08:00	98	114	97	...	0
10	3333	3	16:00	99	116	98	...	0
11	3333	4	00:00	99	130	98	...	0
12	3333	5	08:00	103	141	96	...	1
...	...	...	...	...	...	...	...	...

After Sliding Window

rowIndex	Encounter	Window	Timestamp	Temp (F)	Heart rate	02 count	~300 measures	Outcome
1	1111	[1, 2, 3]	[00:00, 08:00, 16:00]	[98,98,98]	[77,83,86]	[89,89,87]	...	0
2	1111	[2, 3, 4]	[08:00, 16:00, 00:00]	[98,98,98]	[83,86,80]	[89,87,85]	...	1
3	2222	[1, 2, 3]	[00:00, 08:00, 16:00]	[101,101,101]	[88,90,87]	[99,99,99]	...	0
4	3333	[1, 2, 3]	[00:00, 08:00, 16:00]	[98,98,99]	[103,114,116]	[97,97,98]	...	0
5	3333	[2, 3, 4]	[08:00, 16:00, 00:00]	[98,99, 99]	[114,116,130]	[97,98,98]	...	0
6	3333	[3, 4, 5,]	[16:00, 00:00, 08:00]	[99,99,103]	[116,130,141]	[98,98,96]	...	1
...	...	...	...	...	...	...	...	...

(a) Sliding windows example for tabular data

Original Table

rowIndex	Encounter	Window	Timestamp	Notes	Outcome
1	1111	1	00:00	'	0
2	1111	2	08:00	the fox in the box	0
3	1111	3	16:00	said hello to the vox	0
4	1111	4	00:00	'	1
5	2222	1	00:00	the cat said hi	0
6	2222	2	08:00	'	0
7	2222	3	16:00	'	0
8	3333	1	00:00	'	0
9	3333	2	08:00	'	0
10	3333	3	16:00	'	0
11	3333	4	00:00	then the cat said bye	0
12	3333	5	08:00	'	1
...	...	...	...	...	...

After Sliding Window

rowIndex	Encounter	Window	Timestamp	Notes	Notes Sparsity Indicator	Outcome
1	1111	[1, 2, 3]	[00:00, 08:00, 16:00]	the fox in the box said hello to the vox	011	0
2	1111	[2, 3, 4]	[08:00, 16:00, 00:00]	the fox in the box said hello to the vox	110	1
3	2222	[1, 2, 3]	[00:00, 08:00, 16:00]	the cat said hi	100	0
4	3333	[1, 2, 3]	[00:00, 08:00, 16:00]	'	000	0
5	3333	[2, 3, 4]	[08:00, 16:00, 00:00]	then the cat said bye	001	0
6	3333	[3, 4, 5,]	[16:00, 00:00, 08:00]	then the cat said bye	010	1
...	...	...	...	...	...	...

(b) Sliding windows example for nurse notes data

Fig. 1: (a) and (b) represent an example of corresponding tabular and notes data with 24 hour sliding window (length 3). Each rowIndex is an unique 8 hour entry, that belongs to a given Encounter (a patient encounter number), Window length, Timestamp and Outcome (maximum of the original binary Outcome). In (a) around 300 physiological measures were averaged across the 24 hour time interval. And in (b) the Notes in the original table were concatenated within each 24 hour window, with the notes sparsity indicator capturing which window the note came from.

3.1 Data

We use de-identified clinical data from the General Internal Medicine (GIM) ward of a hospital in North America[4], and include patients that have completed visits between December 2014 and April 2019. The data consist of the following sets of features:

- **Tabular data:** This consists of routinely-collected lab results (e.g., blood glucose) and vitals (e.g., hear rate). This data also includes clinical orders, such as diet change orders (e.g., if a patient is moving to a *nil per os*[5] diet) and imaging orders (e.g., chest X-ray).
- **Notes data**: This includes de-identified nursing notes.

We train different models on each set of features (i.e., tabular data vs. notes data) and the models each take as input a window consisting of 3 × 8-hour intervals (i.e., 24 hours of data).

Notes Data Pre-processing To fully utilize the temporal data and provide timely predictions, we concatenate the notes data through sliding windows consisting of 24 hours of data (i.e., 3 × 8-hour intervals of notes). We also notice that sometimes only one note is recorded and appeared in several windows as we slide over the time point when outcome turns 1 from 0, making all those windows share the same content but their corresponding outcomes may vary, thus creating conflicting training pairs with the same input corresponding to different output labels. To deal with this issue caused by the sparsity in the clinical notes, we append a notes sparsity indicator vector of length 3 (i.e., the window size) indicating whether notes were recorded in the corresponding interval, as seen in the Notes Sparsity Indicator in Figure 1b.

Then, for text pre-processing, we remove of white-space, numerical values and standard English stop-words using the `Gensim` library [14]. From the pre-processed notes, we then extract LDA topic probabilities and get the GloVe word representation (more details in the *Models* section).

Tabular Data Pre-processing 8-hour intervals of the tabular data are created by taking the mean average whenever there are multiple measures within an interval. Missing data are imputed with last-observation-carried-forward and then filled in by the mean. For each measure, we then create two additional variables:

- an *indicator variable* for each feature to differentiate between measured and imputed values (i.e., 1 if the value is measured in that interval, 0 if it's imputed); and

[4] A Hospital that is part of Unity Health Toronto.

[5] *Nil per os* is Latin for "nothing by mouth" and is used when a patient cannot receive food orally.

– a *time elapsed variable* for keeping track of the number of hours since the previous measure.

We also use the sliding window technique on the tabular data, we construct the data for each sliding window by concatenating the three observations in a window. (see Figure1a).

To ensure a fair comparison, we remove windows of tabular data where the corresponding notes data window has empty notes. We end up with **133,848**, **22,393**, and **8,631** windows of data for the training, validation, and test sets respectively.

3.2 Models

Notes Data Model We build a CNN model using Global Vectors for Word Representation (GloVe) word embeddings trained on the notes data [12]. The CNN aims to predict patient health outcomes from the nursing notes, which are largely unstructured, free-hand and heterogeneous, with spontaneous entries. GloVe consists of a co-occurrence matrix method based on the idea that words with similar distributions have similar meanings [12]. In early experiments, we found that training GloVe on our own notes yielded better performance.

Each 24-hour interval of nursing notes is treated as a document represented by an array of word embedding vectors. This is given as input to the CNN with 1D convolutions. This is to capture the localized context in the document. We then pair this with the doc2vec rerpesentation of the note to provide more global context on the document level. doc2vec [13] is a document embedding that provides fixed-length dense vector representation for text of variable length and encodes information such as the topic of the paragraph. Overall, our CNN is trained conditioned on both the fixed-length doc2vec vectors and notes sparsity indicators (e.g., $[0, 1, 0]$ indicates notes were recorded only at the second window) to form an ensemble model. See Figure 2 for the architecture of the CNN architecture.

For our GloVe embedding input, we cap the maximum number of tokens to 2,000, and pad with 0's when necessary. This number is determined by the mean average document length in our data. The CNN consists of four 1D convolutional layers, each with double the number of filters as the previous layer. All filters are of size 3, and each convolutional layer is followed by a max pooling layer. We train the doc2vec model (Distributed Memory Model of Paragraph Vectors, PV-DM) on the nurse notes data with 150-dimensional embeddings. We then concatenate the output of the last layer in CNN with the doc2vec representation for the input document to obtain the ensemble model. The concatenated vector is given as input to two densely connected layers, together with notes sparsity indicators.

To reduce overfitting, we apply dropout (45%) before each fully connected layer, and use early stopping when the validation error begins to increase. These parameters are optimized by informal grid search.

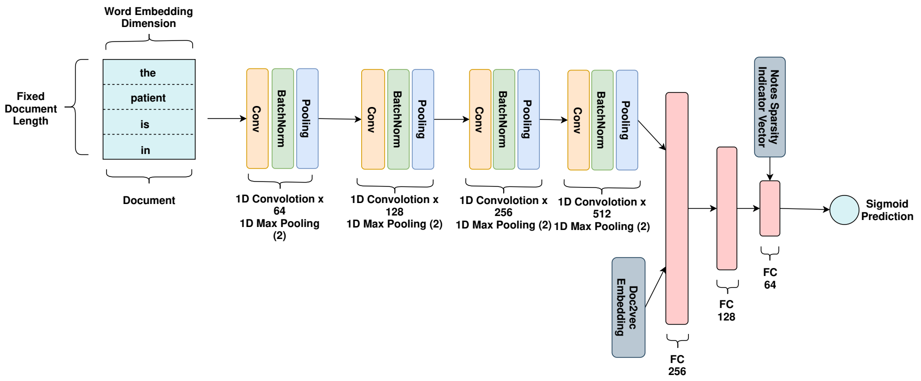


Fig. 2: Model architecture of CNN for unstructured notes data, the inputs of word embedding have fixed length of 2,000 tokens, shorter notes are padded with zeros and longer notes are truncated. Each document consists of notes concatenated from a sliding window. There are four convolution steps before the fully connected layers taking into considerations of the notes level embedding and notes sparsity indicators.

Tabular Data Model In each sliding window of size **3**, there are **375** lab/vital features with **3** temporal observations. In order to leverage the patterns hidden in each feature, we structure the data to have a dimension of **375** × **3** so that each convolutional kernel of size **1** × **3** can focus on a specific feature. During the training processes, the kernels will learn the temporal pattern change across every physiological signal within each sliding window. Extracted features are flattened and fed to two densely connected layers with ReLU activation, followed by a sigmoid layer.

Logistic Regression Models The logistic regression nurse notes model takes as input the topic probabilities from a 50-topic Latent Dirichlet Allocation (LDA) representation of the notes [1]. LDA generates an efficient and low-dimensional representation and offers decent interpretability. The logistic regression tabular data model is trained on **375** × **3** windows of data.

4 Results

4.1 How do our models trained on tabular data compare to models trained on nurse notes data?

For Table 1, the threshold is selected by a grid search to achieve a recall score close to 0.8 that is suggested practically by physicians, which results in a threshold of 0.695 for both the CNN notes model and tabular model.

		OVERALL	**ICU**	**DEATH**	**PAL**
Notes	Accuracy	0.840 (0.831)	0.857 (0.846)	0.859 (0.861)	0.845 (0.850)
	Precision	0.163 (0.195)	0.069 (0.107)	0.015 (0.017)	0.095 (0.094)
	Recall	0.862 (0.807)	0.731 (0.765)	1.000 (0.786)	0.964 (0.867)
	ROC-AUC	**0.858** (0.802)	0.780 (0.774)	**0.984** (0.922)	**0.925** (0.848)
Tabular	Accuracy	0.861 (0.837)	0.882 (0.854)	0.887 (0.869)	0.873 (0.856)
	Precision	0.156 (0.181)	0.069 (0.101)	0.011 (0.019)	0.091 (0.082)
	Recall	0.862 (0.807)	0.769 (0.786)	0.750 (0.929)	0.964 (0.813)
	ROC-AUC	0.839 (0.816)	**0.787** (0.800)	0.908 (0.943)	0.895 (0.843)

Table 1: Overall comparisons between notes model and tabular model and detailed comparisons among ICU transfer, Death and Palliative transfer for the CNN models. The number outside the parenthesis is from the test set whereas the number inside the parenthesis is from the validation set.

The metrics for the CNN models in Table 1 reveal that although both CNN models share the same recall of 0.862, the nursing note CNN model has a higher precision of 0.163 compared to 0.156 from the tabular CNN model, which suggests the former model is more capable of identifying crashing patients, at the cost of lower accuracy, 0.840 as opposed to 0.861. The nursing note data are equally as important, if not more significant, as the tabular data, evidenced by the higher ROC-AUC values on the notes data model. And we observed similar results in the logistic regression models, that the model trained on notes data (ROC-AUC of 0.826) is performing equally as good as the model trained on tabular data (ROC-AUC of 0.820) .

Based on Table 1, among all three individual crashing types (i.e., ICU transfer, palliative transfer, death), ICU transfer has the worst performance. We speculate this is because ICU transfers are inherently highly unpredictable and very spontaneous. We also notice a shift in the precision and recall between the validation and test data, whereby we observed higher recall and lower precision in the test data. We speculate this might be because the distributions in the test and validation data might be different. However, overall, the performance results from the notes and tabular deep models are very similar, with notes performing better on death and palliative conditions.

We also investigate how our deep models capture the true crashing patients. In Figure 3 there are 58 patients who crashed (i.e., who experienced an ICU transfer, a palliative transfer, or death) in our test dataset, out of which, 44 are correctly classified by both of our deep models. And 6 patients are correctly predicted by our notes model only, and another 6 patients are captured by tabular model only. There are only 2 crashing patients who had been missed by both of the deep models.

4.2 How early can our deep models make the correct prediction?

While our deep models are capable of predicting the patients crashing events, we would like to investigate how early before the patient crashing could our deep model signal the warning. The most valuable warning will be those timely before the actual event time. The model can bring assistance to the medical team as resources can be allocated just before the event happens and intervention can be initiated within proper time interval.

For each patient visit, we calculate the time-to-event (in days) from the warning when the model first passes the threshold to the recorded timestamp for the patient crashing. See Figure 4 for examples of predicted risks of crashing by both tabular model and notes model along patient trajectories. The trajectories show how the probability of crashing (y-axis), as predicted by the notes or tabular model, changes over time (x-axis). In the examples, all visits eventually experience the outcome. We highlight the following scenarios:

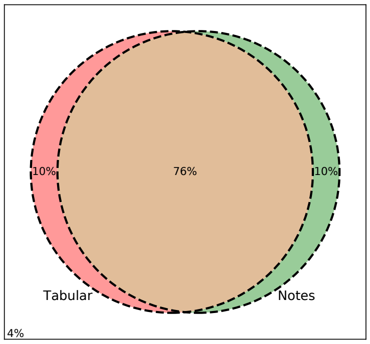


Fig. 3: The pink circle represents the correctly classified patient visits by our tabular model, and the green circle stands for the correctly classified patient visits by our notes model. There are 2 positive patient visits in the test data that are correctly classified by neither of our model, and both of which are ICU transfers patient visits.

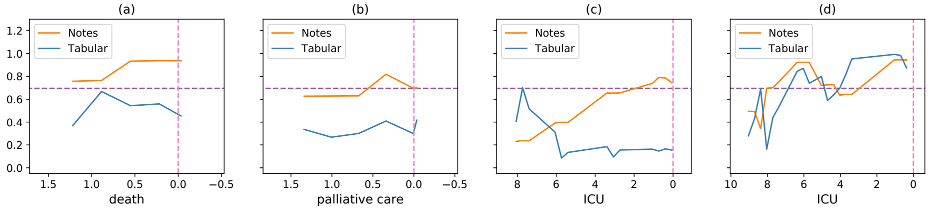


Fig. 4: There are two trajectories in each subplot, one is for the notes model and other one is for the tabular model. The x-axis is right-aligned to the time of event in days. The y-axis is the predictions made by both models. The horizontal line is the threshold used to predict the outcomes. And the vertical line marks the point in which the event happened.

- In Figure 4 (a), the note model predicts that the patient will crash from the beginning when the nurse started to take notes while the tabular model fails to trigger the warning through out the entire patient visit.
- Figure 4 (b) shows the case where probabilities of both models start off below the threshold, but 12 hours prior to the event, the trajectory of the notes

model increases, leading to a successful prediction. The tabular model, on the other hand, never identifies that the patient crashes.

- In Figure 4 (c), we show a case where the tabular model decreased its risk prediction probability, whereas the notes model gradually increased its risk prediction as it approached the time to the event. The probability of the notes model eventually went above the threshold and correctly predicted that the crashing event.
- Finally, in Figure 4 (d), both models output similar risk prediction scores and successfully predict the outcome as time goes close to the crashing event.

We also investigated the time-to-event (in days) for each specific outcome (i.e., ICU transfer, palliative transfer, death). As shown in Table 2, on all three types of crashing, the notes model is capable of making correct classifications with a tighter time to event time than the tabular model, except the ICU transfer at the first quartile.

4.3 What information are the deep models using to make the predictions?

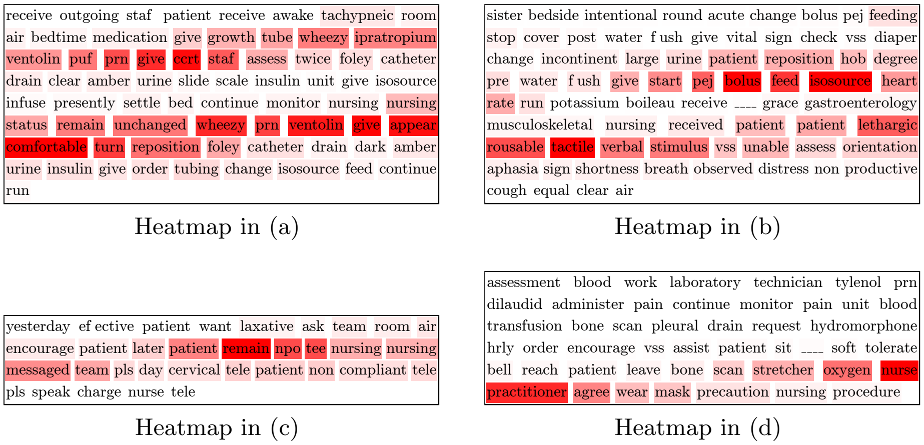
To understand how the deep models make decisions, we adopt the guided grad-CAM technique [15] on both of our models (i.e., tabular model and notes model). This approach calculates the gradients of the risk score with respect to the input: higher gradient magnitude indicates higher activation, which can be used as a measure of how important a given word token is to the prediction. The magnitude of the activation provides a visualization in the form of a heatmap indicating the important tokens contributing to the final prediction.

	Quartile	CRASH	ICU	DEATH	PAL
Notes	0.25	1.102	0.539	2.914	1.413
	0.5	3.536	2.381	11.079	3.817
	0.75	8.203	5.011	19.448	9.391
Tabular	0.25	1.499	0.503	11.913	1.748
	0.5	4.668	3.330	20.347	7.416
	0.75	11.333	6.112	22.049	11.705

Table 2: The time to event (days) for crashing patients visits at three quartiles on the test data set at threshold 0.695.

Guided grad-CAM in the case of 2D image that can be visualized as a 3-channel RGB image. In the case of word embedding with 150 dimensions, it is hard to visualize the guide grad-CAM heatmap. Hence, we take the sum of absolute values along the embedding dimension to represent the magnitude of the activation. This is based on the observation that gradients of the words having high activation will vary dramatically along the embedding dimension. The similar technique can be used in the tabular model to locate important physiological data. We present some interesting cases using the same visits as in Figure 4 in which the notes model provides timely prediction based on meaningful information.

For the case in Figure 4 (a), we show an example heatmap in Figure 5a, that the model focuses on keywords such as **ccrt (critical care response team)** that is terminology related to treatment, and **wheezy**, **ventolin**, and **amber urine** which are symptoms that can give us insights on why the patient died.



Heatmap in (a)

Heatmap in (b)

Heatmap in (c)

Heatmap in (d)

Fig. 5: Heatmap Activation over four notes

Next, we looked at an example of a visit whose probabilities followed different trajectories in the tabular model and in the notes model (i.e., .Figure 4 (b)). We report the corresponding heatmap in Figure 5b, and find that the notes model focuses on words such as **rousable**, **bolus feed** which are typical actions of end-of-life care. This particular visit eventually experienced the palliative transfer outcome. Meanwhile, the tabular model fails to identify the outcome based on the physiological signals.

Then, we look at a case where the tabular data fails to capture the outcome while the notes model does (i.e., Figure 4) (c). In this case, the patient eventually ends up transferring to the ICU. The words with high activation shown in Figure 5c indicate that the patient is under the state of NPO (Latin for “nothing by mouth”) and waiting for a TEE test (i.e., a transesophageal echocardiography).

Finally, in Figure 4 (d), we show an example where the notes model's prediction is in sync with that of tabular model from the beginning of encounter to the crashing time. The notes model identifies the outcome ahead of the tabular model. From the heatmap (see Figure 5d), we find the model is focusing on words like **oxygen**, **mask**, and **precaution**. These words suggest that the patient was undergoing a preemptive act to prevent potential consequence at least noticed by the nurse.

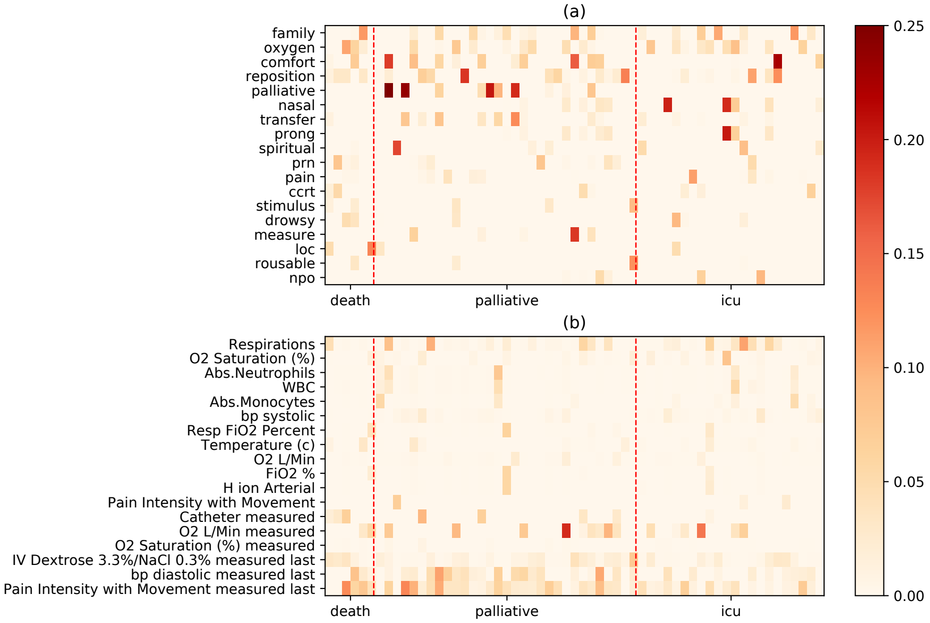


Fig. 6: Important words/tabular features among nursing notes and physiological measurements. x-axis represents sliding windows during the visits of crashing patients (grouped by death, palliative and ICU outcomes), y-axis represents subset of the top indicative words/tabular data as well as those picked from spearman correlation analysis. A physiological data name ending with 'measured' indicates whether the entry is measured or imputed, and 'measured last' indicates the number of hours since last measurement

4.4 What are the top contributing factors of our deep models?

We include a matrix representation of the important words and tabular features from both deep models in Figure 6. Each column represents a sliding window in the test set. In notes model for each word in a sliding window, we add up its magnitude of activation and then normalize it within each sliding window. We then follow the same process to get the matrix representation and manually select top predictive words to be included in the plot.

And for tabular model, we apply the similar process to get top predictive features. And after that we utilize the Spearman correlation analysis to filter out the physiological signals that are highly correlated for a selection of features to be included in the plot.

In Figure 6 (a), we group the sliding windows by outcome type (i.e., ICU transfer, palliative transfer, death). In sliding windows resulting in death, it seems common to have **family**, **comfort**, and **loc (level of consciousness)** mentioned. In visits that result in a palliative transfer, top words are related to the patient being under **measurements**, needing help to keep **comfortable**

(comfort) and getting ready for the **transfer**. Finally, for the outcome of ICU transfer, these sliding windows often involves **nasal**, **prong**, **pain**, and **comfort**.

In Figure 6 (b), the values of some physiological features (e.g., **O2 Saturation**) are more important than whether they are measured or not (e.g., **O2 Saturation measured**). We observe the opposite trend in other features such as **Pain Intensity with Movement**, where the fact that this was measured is more important than the actual value. We also notice that the time since last measurements of certain physiological data usually contributes the most to our model.

5 Conclusion

In summary, we have shown that, although nursing notes are inherently noisy, there is still value in the notes for predicting health outcomes. Deep models trained on nursing notes data can achieve similar performance as those of tabular data. In addition, the notes model is capable of predicting patient crashing closer to the event, ICU transfer, death, or palliative care. When predicting, notes model and tabular model make prediction based on different perspectives. Considering the limitations of the models and potential improvements, we discarded sliding windows if there are no nursing notes data when we build the notes model. To have a fair comparison, we also remove these sliding windows when building the tabular model even if there are other tabular data inside these sliding windows. Future work includes removing high-frequency words in the dataset in the preprocessing step and experimenting with various sliding window sizes or different combination of word embeddings.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
2. Frize, M., Ennett, C.M., Stevenson, M., Trigg, H.C.: Clinical decision support systems for intensive care units: using artificial neural networks. Medical engineering & physics **23**(3), 217–225 (2001)
3. Ghassemi, M., Naumann, T., Doshi-Velez, F., Brimmer, N., Joshi, R., Rumshisky, A., Szolovits, P.: Unfolding physiological state: Mortality modelling in intensive care units. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 75–84. ACM (2014)
4. Ghassemi, M., Naumann, T., Schulam, P., Beam, A.L., Ranganath, R.: Opportunities in machine learning for healthcare. arXiv:1806.00388 [cs, stat] (Jun 2018), arXiv: 1806.00388
5. Ghassemi, M., Pimentel, M.A., Naumann, T., Brennan, T., Clifton, D.A., Szolovits, P., Feng, M.: A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
6. Johnson, A.E., Pollard, T.J., Shen, L., Li-wei, H.L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G.: Mimic-iii, a freely accessible critical care database. Scientific data **3**, 160035 (2016)

7. Khadanga, S., Aggarwal, K., Joty, S., Srivastava, J.: Using clinical notes with time series data for ICU management. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 6433–6438. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1678
8. Khine, A.H., Wettayaprasit, W., Duangsuwan, J.: Ensemble cnn and mlp with nurse notes for intensive care unit mortality. In: 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE). pp. 236–241. IEEE (2019)
9. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/D14-1181
10. Lovelace, J.R., Hurley, N.C., Haimovich, A.D., Mortazavi, B.J.: Explainable prediction of adverse outcomes using clinical notes. arXiv preprint arXiv:1910.14095 (2019)
11. Nguyen, P., Tran, T., Wickramasinghe, N., Venkatesh, S.: `Deepr`: A convolutional net for medical records. IEEE Journal of Biomedical and Health Informatics **21**(1), 22–30 (Jan 2017). https://doi.org/10.1109/JBHI.2016.2633963
12. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (Oct 2014). https://doi.org/10.3115/v1/D14-1162
13. Quoc, L., Tomas, M.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196 (2014)
14. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010)
15. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626 (2017)
16. Smith, G.B., Prytherch, D.R., Meredith, P., Schmidt, P.E., Featherstone, P.I.: The ability of the national early warning score (news) to discriminate patients at risk of early cardiac arrest, unanticipated intensive care unit admission, and death. Resuscitation **84**(4), 465–470 (2013)
17. Waudby-Smith, I.E., Tran, N., Dubin, J.A., Lee, J.: Sentiment in nursing notes as an indicator of out-of-hospital mortality in intensive care patients. PloS one **13**(6), e0198687 (2018)
18. Wellner, B., Grand, J., Canzone, E., Coarr, M., Brady, P.W., Simmons, J., Kirkendall, E., Dean, N., Kleinman, M., Sylvester, P.: Predicting nnplanned transfers to the Intensive Care Unit: A machine learning approach leveraging diverse clinical elements. JMIR Medical Informatics **5**(4) (Nov 2017). https://doi.org/10.2196/medinform.8680
19. Wu, Y., Jiang, M., Xu, J., Zhi, D., Xu, H.: Clinical named entity recognition using deep learning models. In: AMIA Annual Symposium Proceedings. vol. 2017, p. 1812. American Medical Informatics Association (2017)