Detecting Physiological State Changes During Blood Loss via Deep Unsupervised Learning

Chufan Gao[1], Anthony Wertz[1] and Artur Dubrawski[1]

***Abstract*— Monitoring of physiological responses during a blood loss event is crucial in determining appropriate treatment for the well-being of the patient. Physician intuition implies that the body has a number of different physiological response patterns to blood loss, which change as time passes and as blood loss worsens. Although previous research has shown that a random forest classifier is able to determine whether a patient is bleeding based on data alone, it is unclear whether a model is able to detect these accompanying response patterns from raw physiological data. To approach this problem, we use unsupervised machine learning techniques, such as K-means and Agglomerative clustering, as they are designed to extract patterns from data without a ground truth. However, since the data gathered from the patient are high-dimensional and in time series form, it is impractical to handle without further preprocessing. To make this tractable, we employ a deep dilated convolutional encoder with combined with a custom triplet loss function to project the data into a lower dimensional space. By clustering these latent vectors with time constraints and visualizing the clusters over time, we hypothesize that the clusters will correspond to the physiological response patterns that match physician intuition.**

I. Introduction

Internal bleeding is a common symptom from physical traumas, but it is difficult to analyze due to its complexity. The raw data produced by the monitoring equipment is often of multivariate time series form, which is high dimensional and difficult to visually analyze. Machine learning is a natural way of analyzing this high-dimensional data. However, as a whole, the internal bleeding process has not yet been extensively analyzed by the field of machine learning.

This is not to say that no work has been done–for example, Li et al. showed that prediction of whether a crash will happen during blood loss is possible [1]. Falck et al. found that for hemmorhage prediction, a GRU-based model achieves best performance in small false-positive range, while being inferior for negatives compared to a formidable baseline using manually extracted features and a random forest classifier [2]. Lei et al proposed a method of performing supervised classification canonical correlation clusters on time windows of CVP on a pig bleed dataset and was shown to perform well in prediction of bleeding vs non bleeding [3] (the appendix also has a list of medical terms and abbreviations for the reader's convenience). The work previously done on this data has focused on clear-cut, supervised or semi-supervised approaches in validating their hypothesis.

Previous work uses supervised learning to predict clearly delineated outcomes, like the mortality of a person in a given time window. However, there has been a lack of work focusing on less obvious changes in physiological data associated with blood loss. Supervised learning techniques are not applicable to discover these pattern due to the lack of ground truth in analyzing physiological state changes. Only a few studies use unsupervised learning to examine unlabeled data and discover important patterns. Utilizing an unstructured approach may allow us to more extensively understand different physiological effects of blood loss.

Furthermore, Lei et al's work demonstrated that interesting patterns could be found from the clusters - one cluster corresponded mostly to the prebleed phase, a second cluster would take over after bleeding started, and a third cluster would appear even further throughout the bleed. The interpretation was that the physiological responses reflected in the CVP data of the pigs were changing throughout the bleed and that these changes were different physiological responses. Their interpretation was that initially there is an initial compensation reaction to blood loss, and this quickly shifts into a secondary overall systemic reaction. Additionally, they found that most pigs that they analyzed all similarly exhibited such behavior.

This previous work serves as inspiration to our work, as it leads to many additional question concerning the nature of the physiological reactions. For example, are all physiological compensation events universal for each pig? How many such physiological responses are there? Can we build a model to learn and detect these responses?

Despite the widespread adoption of neural networks in data processing, continuous, multivariate time series data have not been affected insofar as, for example, the computer vision or natural language processing communities. The breadth of work in these fields has not translated to this multivariate time series data, even though these fields have valuable models that could be applied. Although deep unsupervised sequence processing techniques have generally focused on natural language processing, many of these model architectures can be generalized to continuous, multi-variable time series data with some additional effort.

[1]Chufan Gao, Anthony Wertz and Artur Dubrawski are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA chufang@andrew.cmu.edu; awertz@cmu.edu; jdolan@awd@cs.cmu.edu

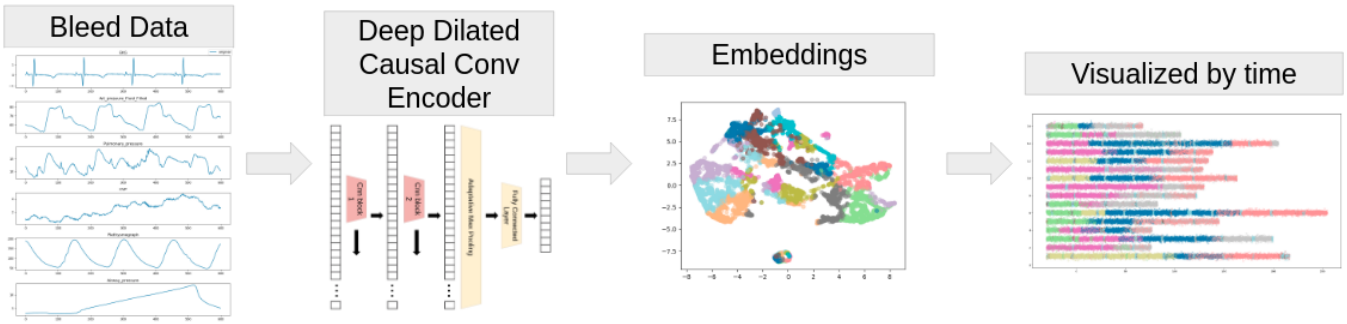


Fig. 1: Overview of the methodology

In this paper, we demonstrate a modern deep unsupervised encoder model in the application of finding embeddings from continuous data of 6 health metrics. With these embeddings, we can then use several different clustering techniques to obtain a number of clusters which may correspond to physician intuition. Additionally, this would expand on Lei et al's findings of physiological response patterns associated with blood loss.

A. Contributions

To summarize, the main contributions of this paper are

1) An investigation as to whether pigs in general have a universal physiological response pattern to internal bleeding.
2) An investigation as to the number of such physiological responses detected from the model.

II. Methodology

An overview is shown in 1

A. Data

The data consists of health metrics of 93 pigs in total. These pigs are separated into 4 groups, which are then bleed at different rates - 60mL/min, 20mL/min, 5mL/min, and 0mL/min respectively. Each pig was monitored for 11 vital signs at 250 Hz (synchronized): arterial and venous blood pressures (CVP, arterial pressure fluid filled and millar, pulmonary pressure), arterial and venous oxygen saturations (SpO2, SvO2), EKG, Plethysmograph, CCO, stroke volume variation (Vigeleo), and airway pressure. The data collection methodology is similar to [4].

For our task, we choose to only use the 16 pigs assigned to the most gradual bleeding task - 5mL/min - as this should give us clearest indications of physiological responses as the pigs' status slowly worsens. Also, we choose to only use 6 of these features - EKG, arterial pressure fluid filled, pulmonary pressure, CVP, plethysmograph, and airway pressure - as they contain potential important semantic information about the physiological status. Additionally, we also have physician annotated timestamps and notes of when a blood draw is performed. This is important as performing a blood draw corresponds with extremely high variation noise in some variables in the time series for a few seconds. Each health metric is measured in 250 hertz.

Since we have 16 pigs and we want to simply analyze all of them, we train our encoder all 16 pigs. This ensures that we are training over all of the data and learning as much from the data as possible. For our training data, we pass in the entire bleed sequence of each pig to the model. To obtain embeddings of the bleed sequences, We choose a window of 600 timesteps and split from the time sequences (without overlap) time windows of 600 timesteps by 6 features. This allows us to get an embedding of the pig state for every 2.4 second window. We chose 600 as it was long enough to get 1 to 2 breaths in and was easier to process computationally; however, this is also a parameter that may be tuned in further research.

B. Causal Dilated Convolutional Neural Network

A convolutional neural network (CNN) is a neural network that trains well on even very high dimensional data, such as images. First introduced in 2012 [5], CNNs have long since been the cornerstone of modern computer vision approaches. However, CNNs can also be applied to non image data as well. For example, although recurrent neural networks have historically been used in sequential data modeling, CNNs are now a popular and viable approach in dealing with sequential data. Wavenet is one of the most famous example of this approach–audio waves produced by the CNN were better than previous state-of-the-art recurrent neural network approaches [6]. Wavenet used dilated convolutions, which skips a timestep every layer of the CNN - leading to an exponentially large effective view of the sequence. It also used causal convolutions, or the idea that the CNN can only process what it has seen before in the time sequence. In our application, causal convolutions are still important because it allows the model to be run on arbitrary length sequences and also be able to run on sequences online during test time. Should we extend our approach to online detection of bleeding responses phases, our model wouldn't break.

For our deep unsupervised embedding model we use an convolutional encoder as proposed by Franceschi et

al. [7]. Compared to an autoencoder, training solely an encoder model is beneficial in that it severely reduces the training time necessary for the model to achieve good performance; additionally, in many cases, training a decoder model is unnecessary to obtain meaningful embeddings, as shown by Franceschi et al. This model is able to produce meaningful embeddings that perform close to state of the art if not better compared to even supervised time series embedding methods [7]. Using CNNs to process sequences have also another inherent advantage: the speed. Since CNN operations are highly parallelizable, training this encoder is faster than training a traditional sequence to sequence recurrent neural network [7]. We choose to use this model as opposed to traditional statistical feature extraction as we want to assume as little as possible about the data and see if the network can discover patterns by itself. Further information about the nature of this convolutional structure is shown in Figure 2.

While only 3 dilated causal CNN layers shown here, in the actual model, the number of these modules is a hyperparameter that we can specify. The dilated part is visualized by the doubling of the gaps between the boxes that is analyzed by the CNN as you go further up in the output layers. The causal part is shown by the fact that the final output box on the upper right only has access to the information of the boxes before it–a regular CNN would appear more symmetric in that it have access to information after its current timestep as well. The adaptive pooling layer comes after the dilated causal CNN layers and simply reduces an arbitrary dimensional input to a fixed-sized output. Finally, the final layer is a fully connected layer that outputs the embedding.

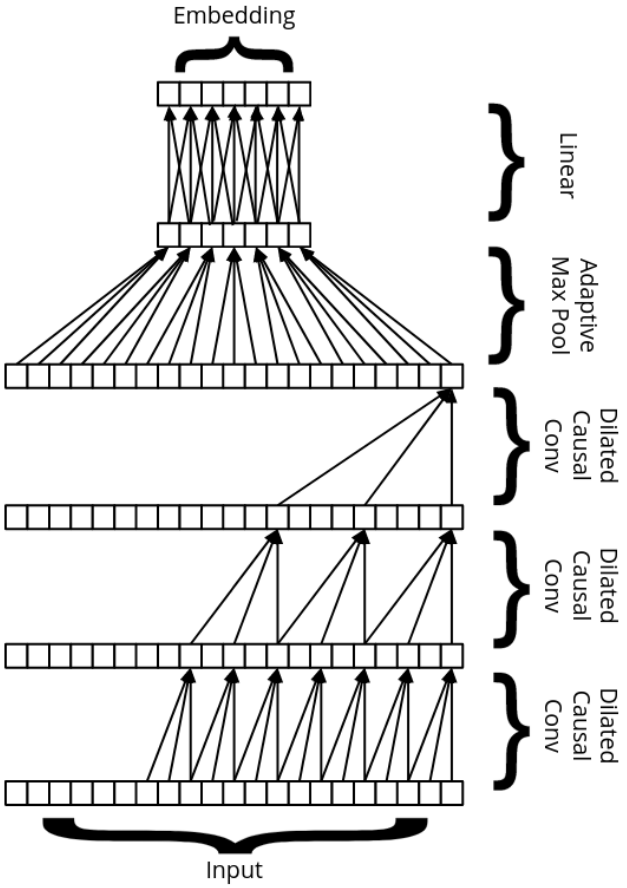


Fig. 2: Graphical representation of dilated and causal convolutions. The output of each row is the row above it. The bottom-most row of boxes denotes a variable length input, and the top-most row of boxes an out of latent vector embeddings.

C. Triplet Loss

We will use triplet loss to train our encoder as specified by [7] et al. Triplet loss is a loss function with a very natural intuition as its basis - similar things should be close together and unsimilar things should be further apart. This is reflected in its mathematical formulation. Let f be our encoder that obtains latent vectors from the time series data. Let x, x^{pos}, x_k^{neg} be the reference time series, a positive time series example, and a negative time series example. Let K be the number of negative samples to take. Then, the loss is shown in equation 1.

$$L = -log(\sigma(f(x)^T f(x^{pos}))) - \sum_{k=1}^{K} log(\sigma(-f(x)^T f(x_k^{neg}))) \quad (1)$$

Triplet loss is popular in natural language processing - Word2vec [8] as it is effective in training unsupervised models that obtain latent vectors from words that encode some semantic meaning. Franceschi et al. demonstrated that this is useful for unsupervised learning of useful embeddings of general multivariate time sequences as well [7].

D. Sampling methodology

We use a modified version of Franceschi et al.'s sampling algorithm to obtain choices of reference x, positive example x^{pos}, and negative example x_k^{neg}. This is different from the original implementation from Franceschi et al. since the negative samples are only choosen randomly when there can be no overlap with the reference time series; thus, this guarantees that a negative example can't be a positive example as well. This should allow the model to learn better as there is a clearer difference between positive and negative samples. Algorithm 1 shows the methodology. Figure 3 shows how our proposed sampling algorithm in practice.

E. Evaluation

Since we have no ground truth, validation of usefulness of these embeddings is an open problem. However, we can qualitatively evaluate them. We use a variety of different clustering methods to try to find the one that produces the best cluster of embeddings that makes sense to us intuitively. We are looking for is separation of the clusters by time - that is - different clusters should be separated

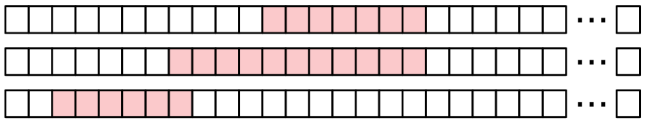


Fig. 3: Sampling methodology. Let the each row of boxes represent multivariate sequences of our training data, and let the rows be the sequences in the batch where we choose the references from. Then, the pink squares represent the reference timesteps, positive timesteps are sampled form within the reference timesteps. Negative timesteps are sample from the white boxes (all of the timesteps excluding the reference timesteps).

for $i \in [1, N]$ **do**
  randomly choose length of reference sample $len_i^{ref} \in [1, len(y_i)]$
  randomly choose length of positive sample $len_i^{pos} \in [1, len_i^{ref}]$
  randomly choose reference sample x_i^{ref} from subseries of y_i of length len_i^{ref}
  randomly choose positive sample x_i^{pos} from subseries of x_i^{ref} of length len_i^{pos}
end for
for $k \in [1, K * N]$ **do**
  randomly choose y_k from the batch
  randomly choose $len_k^{neg} \in [1, size(y_k)]$
  Let $x_{y_k}^{ref}$ be the reference sample that we previously took from y_k
  randomly choose x_k^{neg} among subseries of y_k of length len_k^{neg} without overlapping $x_{y_k}^{ref}$
end for

Algorithm 1: Input: A training batch of complete sequences y_i, where i is the ith training sequence. Let N be the total number of sequences in this training batch. Let K be the ratio of negative samples to be sampled per batch item. Output: N reference samples x^{ref}, N samples of positive samples x^{pos}, and N*K negative samples x^{neg}.

from each other by time, but the same cluster should not be separated by time. Another thing that we are looking at is order consistency of the clusters over time. For example, since the pigs are only ever getting worse in our data, and should never return to a previous "healthy" state.

Specifically, we use a number of different time embeddings, clustering techniques, and latent dimensions, and finally, number of clusters. We explore:

1) Time embeddings added to latent embeddings. The type of time embedding is taken from attention transfomers, from Vaswani et al. [9].
   a) No time information added
   b) Adding time information for full length of the sequence
   c) Adding time information only from the start of bleed (Since we know the exact location of the start of bleed from the physician annotations, we only add temporal information to the embeddings obtained after the pig starts bleeding. The prebleed embeddings are left alone. This is effectively adding information about the amount of blood lost, since bleed speed is constant after the pig starts bleeding).
2) Clustering methods (All of these are implemented in sklearn [10])
   a) K-means
   b) Agglomerative clustering with ward linkage (Bottom-up hierarchical clustering. Ward's linkage merges the two clusters such that the increase in the value of the sum-of-squares variance is minimized [11]. Specifically, sklearn references [12].)
3) Latent embedding dimensions: 64, 128, 256
4) Number of clusters: In addition to all of these methods, we also explore 11 different numbers of clusters that we pass into the clustering algorithms (from 2 to 12 clusters).

III. Results

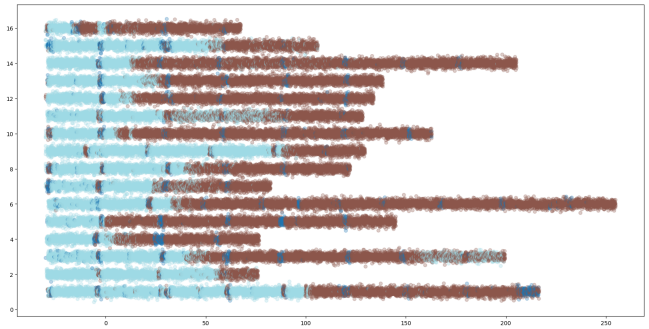
A. Graphs of the clusters

Since there may be 3 clustering methods * 3 time embedding methods * 3 latent embedding dimensions * 11 number of clusters = 297 possible graphs in total, it is impractical to show all of them in this paper. Thus, we only show a few examples shown in Figure 4. However, all of the figures and model code (along with hyperparameters) will be on my github: github.com/andy1445. The Y axis of each of the following figures represents each pig, and the X axis represents minutes. The different colors represent different clusters found by the clustering algorithm, and the colors are random. To make these plots easier to see, we jittered the Y axis.

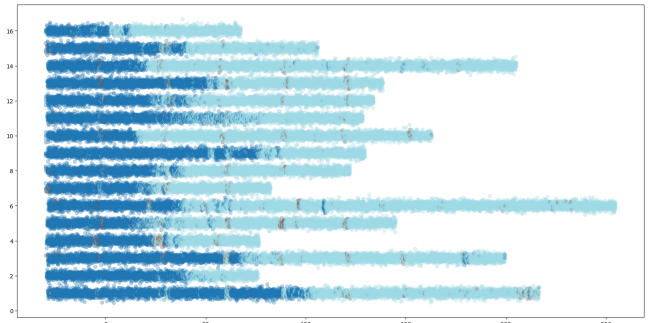
IV. Discussion

From these graphs, we can see that the clustering algorithm is at able to discern a "healthy" state and a "non healthy" state for all of the 3 cluster graphs. For the 11 cluster graphs, for both kmeans and agglomerative clustering we are able to discern between different pigs as well as different cluster progressions throughout the bleed; however, all pigs eventually end up in the same state - the green or purple cluster for kmeans and the orange cluster for agglomerative clustering. This make sense as the pigs all crash after this last cluster. Additionally, we found that reactions between pigs are not universal, as some pigs skip clusters entirely through the bleed, and they can also start off in different clusters compared to other pigs. We also see that pigs can go through as many as 5 different states and as low as 2 states. Additionally, we are able to detect bleed draws. They show up as noise that occurs regularly every few 30 minutes or so in the plot of the latent clusters.

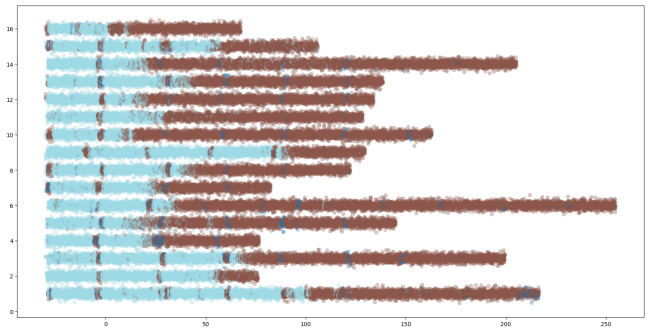
Fig. 4: The plots of the clusters over time when the model is trained with the ability to differentiate between differences between individual pigs. The colors of the clusters are not consistent between the plots. The X axis is in minutes, and the Y axis are the pigids.



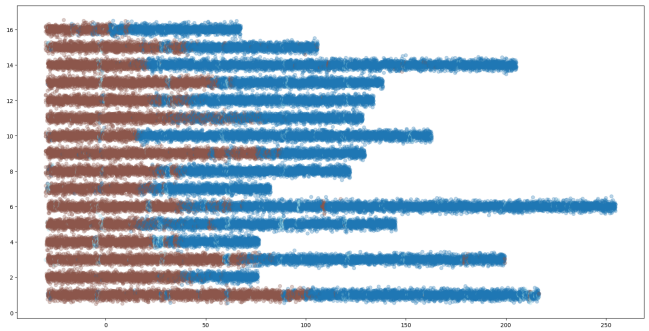
(a) 128 dimensional latent vectors, half time embedding, agglomerative clustering, 3 clusters



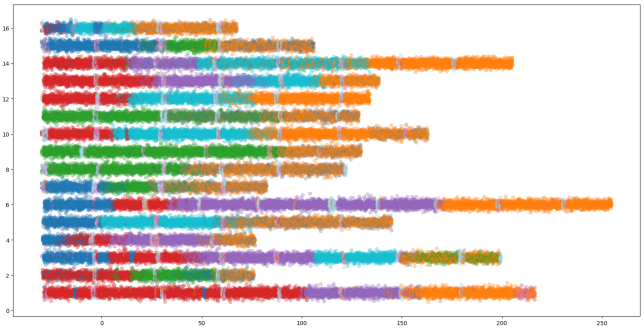
(b) 128 dimensional latent vectors, half time embedding, kmeans clustering, 3 clusters



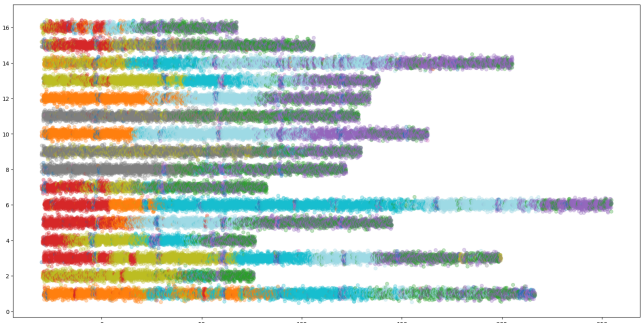
(c) 128 dimensional latent vectors, no time embedding, agglomerative clustering, 3 clusters



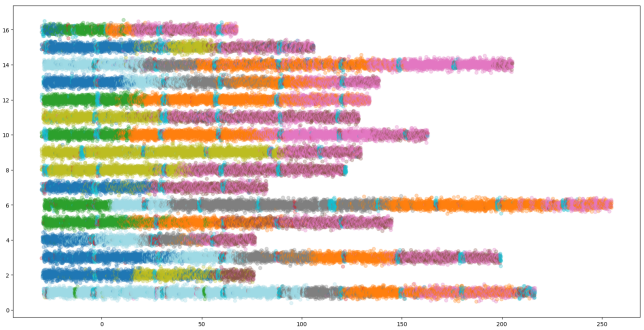
(d) 128 dimensional latent vectors, no time embedding, kmeans clustering, 3 clusters



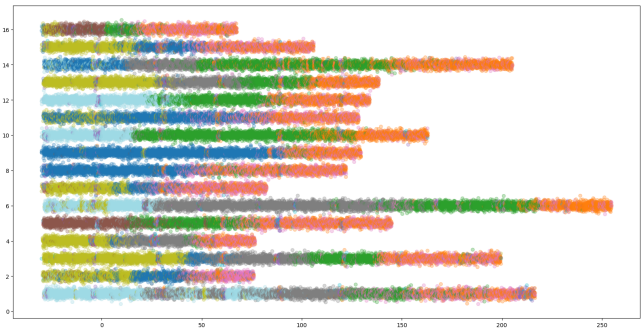
(e) 128 dimensional latent vectors, half time embedding, agglomerative clustering, 11 clusters



(f) 128 dimensional latent vectors, half time embedding, kmeans clustering, 11 clusters



(g) 128 dimensional latent vectors, no time embedding, agglomerative clustering, 11 clusters



(h) 128 dimensional latent vectors, no time embedding, kmeans clustering, 11 clusters

V. Future Work

Future work should seek to use more rigorous evaluation metrics. Once we have the labels, we can then train a random forest classifier to predict the labels from the raw embeddings alone. Additionally, we should also use our embeddings to predict bleed or survival, like previous work has done. Future work should also explore more variety of models such as different encoder models such as Variational Autoencoders. Additionally, Additionally, we can also find clusters or separations in a multitude of different ways, for example, we can use Hidden Markov Models (HMMs) to detect changes as well as change point detection. We should also aim to have physicians analyze the validity of the different clusters that we find.

VI. Acknowledgements

The authors would like to acknowledge the Robotics Institute Summer Scholar program for facilitating this research. We'd also like to acknowledge the Auton Lab, for the invaluable guidance from its members throughout the summer.

Appendix

Abbreviation and Definitions

- Airway pressure: Pressure in the the airways, note that this is under artificial ventilation to offset the energy requirement for breathing, so that a cleaner reaction to the blood loss may be obtained (measured in mmHg) [2].
- Arterial pressure fluid filled: Systemic arterial blood pressure in the aorta, measure in mmHg [2].
- Arterial pressure millar: Systemic arterial oxygenated blood pressure in the peripheral (measured in mmHg) [2].
- CCO: Continuous cardiac output is a measure of the volume of blood pumped from the heart in a certain amount of time. It is often used as a predictor of oxygen delivery to the cells (measured in mL/s) [13]
- CVP: Central venous pressure is a measure of pressure in the superior vena cava that can be used as an estimation of right atrial pressure, often used as an assessment of hemodynamics and hemmorage prediction, particularly in intensive care units (measured in mmHg) [14].
- EKG/ECG: Both are the exactly the same thing and stand for electrocardiogram, which is a measure of the flow of the cardiac electrical cycle (measured in mV) [15].
- Plethysmograph: A waveform that represents changes in blood volume. It has no units and is qualitative due to the non-linear relationship between the absorption of the light for each individual, but overall patterns can be [16].
- Pulmonary pressure: Pulmonary artery pressure (PAP). The pressure of blood pumped from heart into pulmonary (lung) system (measured in mmHg) (i.e. deoxygenated) [2].
- SpO_2: Arterial oxygen saturation (oxygenated) (measured in %) [2].
- SvO_2: Venous oxygen saturation (deoxygenated), (measured in %) [2].
- Vigeleo: Variation of the stroke volume (SV) or the volume of blood pumped out of the left ventricle (oxygenated), defined as range over mean [2].

References

[1] X. Li, M. R. Pinsky, G. Clermont, and A. Dubrawski, "Leveraging routine pre-operative blood draws to predict hemorrhagic shock during surgery," in *Proceedings of the Neural Information Processing Systems Machine Learning for Health (ML4H) Workshop 2018*, 2018.

[2] F. Falck, M. R. Pinsky, and A. Dubrawski, "Deep sequence modeling for hemorrhage diagnosis," in *Proceedings of the Neural Information Processing Systems Machine Learning for Health (ML4H) Workshop 2018*, 2018.

[3] E. Lei, K. Miller, and A. Dubrawski, "Learning mixtures of multi-output regression models by correlation clustering for multi-view data," *arXiv preprint arXiv:1709.05602*, 2017.

[4] M. R. Pinsky, "Instantaneous venous return curves in an intact canine preparation," *Journal of Applied Physiology*, vol. 56, no. 3, pp. 765–771, 1984.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[7] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," *arXiv preprint arXiv:1901.10738*, 2019.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] T. W. Liao, "Clustering of time series dataa survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[12] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[13] J.-L. Vincent, "Understanding cardiac output," *Critical care*, vol. 12, no. 4, p. 174, 2008.

[14] P. Shah and M. A. Louis, "Physiology, central venous pressure," in *StatPearls [Internet]*. StatPearls Publishing, 2018.

[15] M. B. Conover, *Understanding electrocardiography*. Elsevier Health Sciences, 2002.

[16] J. Nijboer, J. Dorlas, and H. Mahieu, "Photoelectric plethysmography-some fundamental aspects of the reflection and transmission methods," *Clinical Physics and Physiological Measurement*, vol. 2, no. 3, p. 205, 1981.