Longitudinal Predictions on ICU Data

Report Presentation at https://youtu.be/9HCeKJVk_vc

Greg Harrison
gdh3@illinois.edu

***Abstract*—This project involves assessing ways to make predictions on patient mortality based on the care they receive when they are admitted to the intensive care unit (ICU). Data from the MIMIC-III database was used to train a long short-term memory network (LSTM) and a feed forward neural network to predict whether a patient will survive more than one year after they were admitted to the ICU. The MIMIC-III database contains deidentified patient healthcare data gathered from around 60,000 ICU admissions. To create timeseries of data to train the model, data from MIMIC-III that is associated with a timestamped encounter between a patient and a healthcare professional was aggregate into "care events", which are a list of all healthcare activities performed on the patient in a defined increment of time. To find the optimal number of care events per admission to train the model with and use it for making mortality predictions these deep learning models and were created using various numbers of care events per admission. It was determined that the LSTM model is better for predicting the mortality of patients that have more than 30 care events per admission, and the feed-forward neural network is better at predicting outcomes of patients with fewer than 30. Additionally, an autoencoder was used to create an embedding from the care events. T-Distributed Stochastic Neighbor Embedding (t-SNE) was used to reduce the dimensionality of the embedding so that it can be visualized. A visualization of the embedding, mapped to patients that survived more than a year after ICU admission and those that did not, show distinct clustering of data for both classes, indicating that the autoencoder produced a meaningful embedding. These embeddings were used to train the LSTM and feed-forward neural network models. When compared to the models trained using one-hot vector representations of care events, the models trained with the autoencoder embeddings performed better for the LSTM model and feed forward neural network.**

***Index Terms*—Deep learning, Machine learning, MIMIC-III, Long Short-Term Memory Network, LSTM, t-Distributes Stochastic Neighbor Embedding, t-SNE**

1. INTRODUCTION

The goal of this project is to determine whether the care a patient received during their visit to the ICU is a good predictor of patient mortality. This is an important area of study, because being able to make mortality predictions can help hospitals determine which patients are most at risk so they can decide how best to allocate resources.

The data for this project is from the Medical Information Mart for Intensive Care (MIMIC-III) database. This database contains data on patient visits to the intensive care units at Beth Israel Deaconess Medical Center [1]. It contains data such as when a patient was admitted and when they were released, what the patient was diagnosed with, what procedures were performed, what medications were prescribed, and lab results gathered.

This project is modeled off the research of Beaulieu-Jones, Orzechowski, and Moore described in *Mapping Patient Trajectories using Longitudinal Extraction and Deep Learning in the MIMIC-III Critical Care Database* [2]. This paper compared different methods for predicting whether a patient would survive more than one year past their initial admission date to the ICU. They compared five models for making this classification: a random forest model, a logistic regression model, a support vector machine, a feed forward neural network, and a long short-term memory network (LSTM) [11]. They organized the data into care events, which were defined by creating a sequence of all actions initiated by a healthcare provider after a patient was admitted to the ICU and grouping all actions that occurred in sequence that had 59-minutes or less time between actions, viz., sequential events that occurred more than 59-minutes apart would be part of different care events. The 59-minute time interval was determined through experimentation. To train the model on the traditional machine learning methods they created a snapshot vector by averaging values from a set of care events. They compared these methods using the area under the receiver operator characteristic curve and found that the LSTM model was best at predicting patient mortality.

Additionally, Beaulieu-Jones et al. [2] built off the work of [3, 4, and 5] by exploring the use of a stacked autoencoder to create embeddings for care events. They use t-Distributed Stochastic Neighbor Embedding (t-SNE) [6] to visualize this. The visualization showed distinguishable clusters of both classes, i.e., patients that survived more than one year after admission and patients that did not. This proved that the autoencoder was able to produce a meaningful embedding.

This project attempts to replicate the LSTM models and the autoencoder embeddings produced by Beaulieu-Jones et al. Additionally, this project explores how the autoencoder is producing an encoding, by using t-SNE to visualize the hidden states at each layer of the encoder portion of the autoencoder. This exploration is modeled off the work of Yu et al. in their paper, *Monitoring ICU Mortality Risk with A Long Short-Term Memory Recurrent Neural Network* [7]. In this paper the authors described using t-SNE to create a visualization of the hidden states of their LSTM model at each timestep. In these visualizations, they plotted the instances of patient deaths and instances of patients that survived with different colors. At the first timestep these two classes are completely intertwined; by the final timestep the two classes are separated into unique clusters. By doing this they were able to show that their model was creating better embeddings with each timestep.

2. METHOD

2.1 Data Processing

A list of patient care events that occurred in each visit for each patient, and that have associated timestamp information to tell when an event occurred, was compiled using data from the INPUTEVENTS_CV, INPUTEVENTS_MV, LABEVENTS, DATETIMEEVENTS, and PROCEDUREEVENTS_MV tables of the MIMIC-III database. INPUTEVENTS_CV contains data from the CareVue database [9] on fluids that were given to patients, timestamped by when the event was charted. The INPUTEVENTS_MV contains data from the MetaVision [8] database on fluids that were given to patients, timestamped by when the event began and when it ended. LABEVENTS contains lab test data, timestamped by when the sample was taken from the patient. DATETIMEEVENTS contains all events that have an associated datetime, timestamped by when the event was charted. PROCEDUREEVENTS_MV contains the procedures that a patient underwent, timestamped by when the procedure started and when the procedure ended.

The care events for each patient visit were sorted by their timestamps. Care event IDs were given to each event, such that all events that occurred less than or equal to 59-minutes apart would share the same care event ID.

Additional patient information including date of birth, date of death (for deceased patients), admission location, insurance provider, patient’s religion, language, ethnicity, and marital status was gathered from the MIMIC-III ADMISSIONS and PATIENTS tables.

All events that shared the same care event ID were combined into one row such that each care event had one row in the dataset that would describe all actions that were performed by healthcare officials within that window of time. These care events are associated with admission IDs, so that when looking at a single admission, you can see all the care events that happened during that patient admission, and what occurred during the span of time that made up each care event.

2.2 Creating Embedding with Autoencoder

An autoencoder was used to create an embedding of the care event data. The autoencoder consisted of eleven layers: six layers for encoding the data to the latent embedding and five layers to reconstruct the data. The encoder had 1722, 1024, 512, 256, 128, 64 nodes in each respective layer. The decoder had 128, 256, 512, 1024, and 1722 nodes in each respective layer. Dropout of 20% was used between the input layer and the first hidden layer. A sigmoid activation function was used at each hidden layer. The optimizer used was ADADELTA [10] and the loss function used was binary cross-entropy.

2.3 Predicting Survival One Year After Admission

2.3.1 LSTM

An LSTM model was trained to make predictions on whether a patient will survive more than one year after being admitted to the ICU. The model was trained using 5-fold cross validation. The area under the receiver operator characteristic curve (AUC ROC) was the metric used to score the model. The model was comprised of an embedding layer, followed by three LSTM layers, and ending with a fully connect layer. The first LSTM layer had 100 nodes, and the second and third LSTM layers had 50 nodes each.

To see how the number of care events per admission impacts the predictive capability of an LSTM model, eight LSTM models were trained using 1, 3, 5, 10, 20, 30, 50 and 70 care events per admission, respectively, to train each model.

To do this, the care events for each admission were combined into a list. If more than the desired number of care events occurred in a single visit, the number of care events associated with that visit were truncated to the desired amount of care events. If fewer than the desired number of care events occurred in a visit, the list of care events was padded with care events that contained 0 for all variables up to the desired number.

For example, let us look at three admissions, and assume that admission 1 has 2 care events, admission 2 has 5 care events, and admission 3 has 4 care events. The model will be trained using these admissions, which are represented as a list of care events, which are themselves encoded lists representing all the encounters between a patient and a health care provider during that care event. If we were to train the model with these three hypothetical admissions, using a maximum of 3 care events per admission, these three admissions will look like the following, where CE is a care event, and PADDING is list of length CE filled with 0s.

Admission 1: [CE, CE, PADDING]
Admission 2: [CE, CE, CE]
Admission 3: [CE, CE, CE]

When the model is trained using a maximum of 5 care events per admission, these three admissions will be represented as follows:

Admission 1: [CE, CE, PADDING, PADDING, PADDING]
Admission 2: [CE, CE, CE, CE, CE]
Admission 3: [CE, CE, CE, CE, PADDING]

The same LSTM model was then trained using the latent embedding from the autoencoder instead of the one-hot encoded care event vectors.

2.3.2 Feed Forward Neural Network

A feed forward neural network was also trained to predict patient mortality one year after admission. The model consisted of an input layer, three hidden layers of 300 nodes each, and an output layer. Dropout of 50% was used between the input layer and first hidden layer, and dropout of 20% was used between the last hidden layer and the output layer. This model was trained using 5-fold cross validation. It was scored using AUC ROC to compare with the LSTM models.

Like the LSTM model, this model was also trained using 1, 3, 5, 10, 20, 30, 50 and 70 care events per admission using both the one-hot encoded care events and the latent embedding of care events from the autoencoder.

3. RESULTS

3.1 *Care Events*

1721 unique types of encounters between patients and healthcare workers that had a timestamp associated with them were found to exist in the MIMIC-III data used. This means that after one hot encoding was performed on the care events, each care event was a list of length 1721.

Table 1 shows a statistical description of the dataset after care events were aggregated for each patient admission. Figure 1 shows the distribution of the number of care events per admission.

Table 1. Statistical summary of the data used.

Number of admissions	7,788
Number of care events	559,035
Mean number of care events per admission	71.782
Care events per admission standard deviation	127.33
Minimum number of care events per admission	1
Maximum number of care events per admission	2624
Median number of care events per admission	34
Patients survived more than 1-year after admission	72.70%

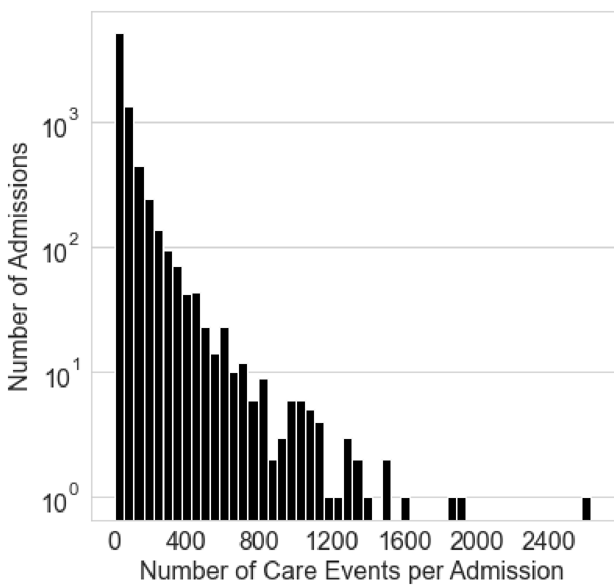

Figure 1. Histogram showing the number of care events per admission.

3.2 *Autoencoder Embeddings*

To see if the autoencoder was creating meaningful embeddings of the care events, t-SNE was used to visualize the embeddings. Figures 2 – 4 show visualizations created using t-SNE on the input layer and hidden layers of the autoencoder. Each point on the plots is labeled by whether the patient survived more than one year after being admitted to the ICU or if they did not. By looking at the visualization from each layer, we can see how the encoder starts to form embeddings. In the input layer (Figure 2), some small clusters can be seen, but there is a significant dispersion of points in both classes throughout the entire plot. In contrast, the plot of the 5th layer (Figures 4) displays larger, more recognizable clusters, of both classes. This indicates that the autoencoder is likely producing meaningful embeddings of the care event data. However, it is hard to draw significant conclusions about the efficacy of the autoencoder embeddings to improve model performance given the significant amount of point dispersion in the t-SNE plot of the latent embedding (Figure 4).

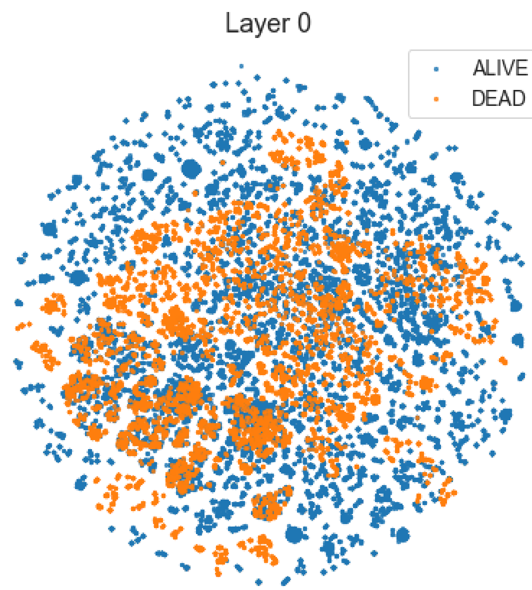

Figure 2. Plot created from performing t-SNE on the input layer of the autoencoder. Uses embedding for 50,000 admissions and shows which patients survived more than one year after the admission date.

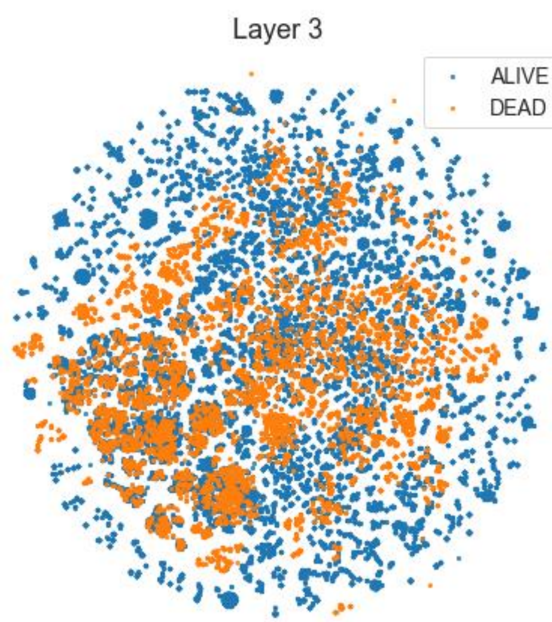

Figure 3. Plot created from performing t-SNE on the 3rd layer of the autoencoder.

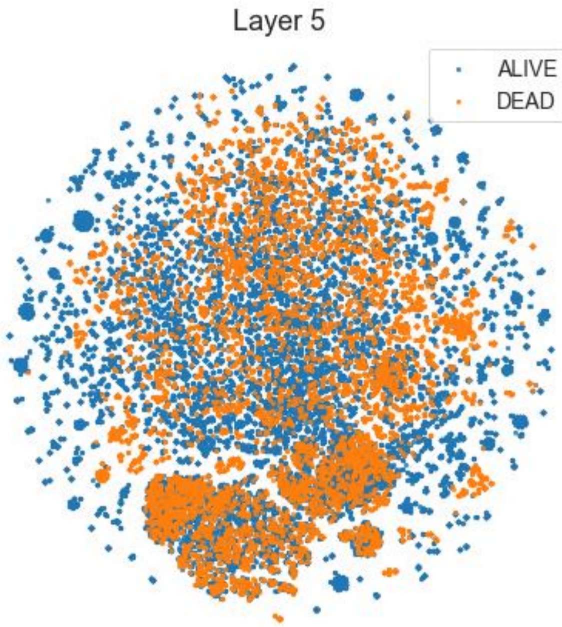

Figure 4. Plot created from performing t-SNE on the 5th layer of the autoencoder. This is the latent embedding.

3.3 Mortality Prediction

3.3.1 LSTM using One-Hot Vectors

Figure 5 shows the AUC ROC results for the LSTM model when different numbers of care events per admission were used for training. Each boxplot is made from five AUC ROC scores, viz., one for each fold of the 5-fold cross validation. These models used the one-hot vector representations of the care events for training.

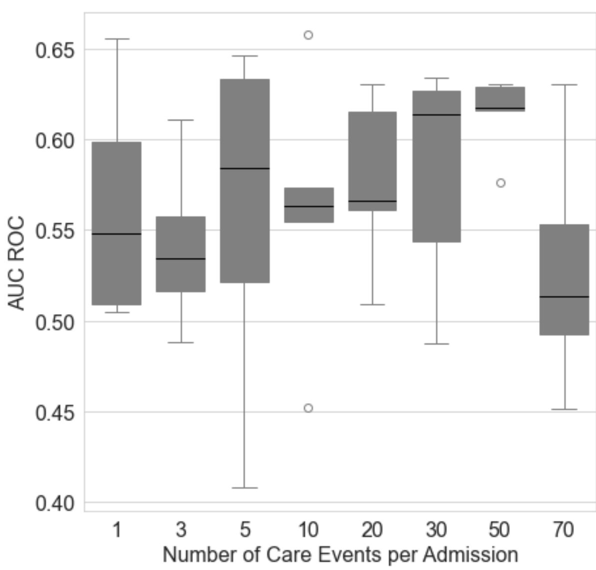

Figure 5. Comparing results of 5-fold cross validation of LSTM trained using 1, 3, 5, 10, 20, 30, 50, and 70 one-hot encoded care events per admission.

3.3.2 LSTM using Autoencoder Latent Embedding

Figure 6 shows the AUC ROC results for the LSTM model when different numbers of care events per admission were used for training. Each boxplot is made from five AUC ROC score, viz., one for each fold of the 5-fold cross validation. These models were trained using the autoencoder embedding vector representation for care events.

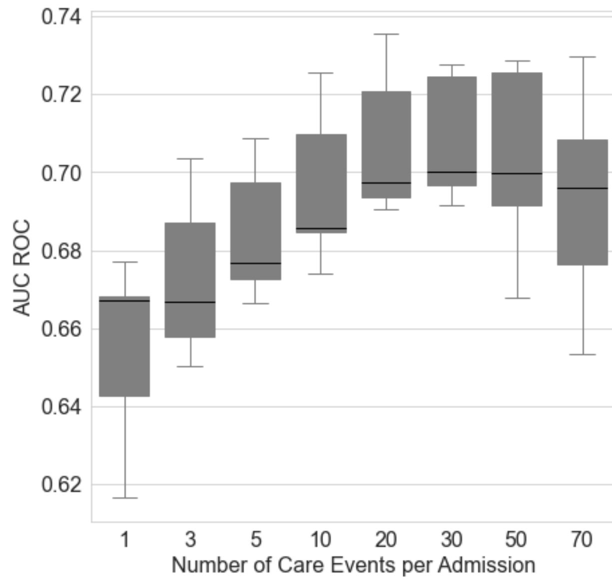

Figure 6. Comparing results of 5-fold cross validation of LSTM trained using 1, 3, 5, 10, 20, 30, 50, and 70 care events per admission, where care events were represented using the latent embedding vector representation from the autoencoder.

3.3.3 Feed-Forward Neural Network using One-Hot Vectors

Figure 7 shows the AUC ROC results for the feed-forward neural network model when different numbers of care events per admission were used for training. Each boxplot is made from five AUC ROC scores, viz., one for each fold of the 5-fold cross validation. These models used the one-hot vector representations of the care events for training.

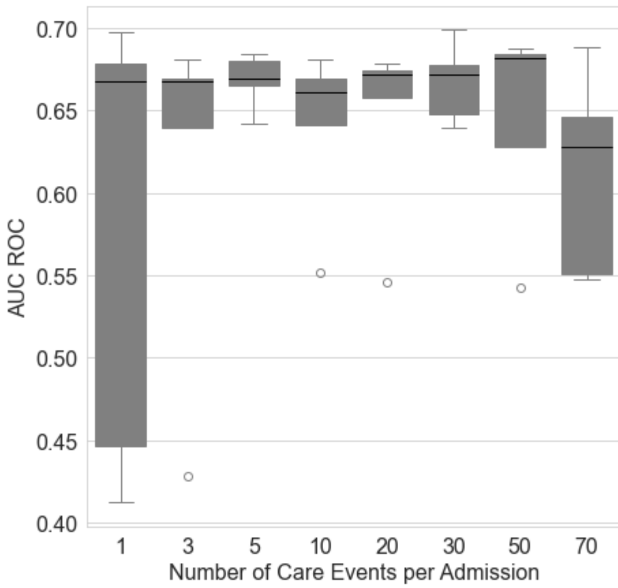

Figure 7. Comparing results of 5-fold cross validation of feed-forward neural network trained using 1, 3, 5, 10, 20, 30, 50, and 70 one-hot encoded care events per admission.

3.3.4 Feed-Forward Neural Network using Autoencoder Latent Embedding

Figure 8 shows the AUC ROC results for the feed-forward neural network model when different numbers of care events per admission were used for training. Each boxplot is made from five AUC ROC score, viz., one for each fold of the 5-fold cross validation. These models were trained using the autoencoder embedding vector representation for care events.

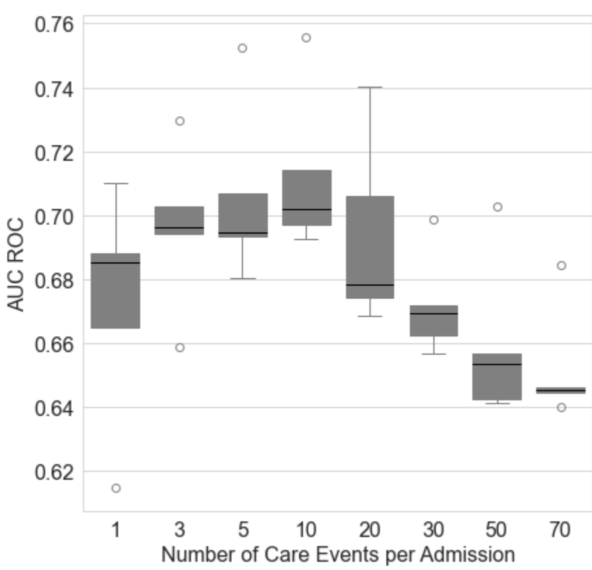

Figure 8. Comparing results of 5-fold cross validation of feed-forward neural network trained using 1, 3, 5, 10, 20, 30, 50, and 70 care events per admission, where care events were represented using the latent embedding vector representation from the autoencoder.

4. CONCLUSION

The LSTM models perform better than the feed forward neural network models when more care events are used. This indicates that an LSTM model may be useful for predicting mortality of patients with conditions that require them to have long hospital stays. However, when a patient has around 20 care events or fewer over the course of their ICU admission, it would be better to use a feed-forward neural network model to predict the patient's outcome.

The AUC ROC results for the LSTM model trained using the autoencoder embeddings indicate that the optimal amount of care events to use for training the model and making predictions on patient mortality is 30 or more. This is likely because the median number of care events per admission is 34. The model performance begins to suffer as the number of care events used exceeds 50. This is likely because more admissions are having to uses padded care events as the number of care events used increases.

From looking at the point clusters in the t-SNE plots of the autoencoder embedding (Figure 4), it appears that the autoencoder is creating a meaningful embedding representation of the care events. The viability of these embeddings in training models to predict patient mortality can be seen in Figure 9, which compares the results of models that were trained using the autoencoder embedding vectors and those that used the one-hot vectors. In all cases, the models trained using the autoencoder embeddings outperforms the models trained using the one-hot vectors, regardless of the number of care events per admission used to train the model.

Future research could be performed to prove the theory that the LSTM model performs better on admissions with many care events by removing admissions that had care events below a certain threshold amount and measuring the performance of the model on this reduced dataset. To go a step further, research could be done to see what types of admissions tend to have fewer care events and removing those ones. For instance, maybe fractured bones have fewer care events on average and all admissions that involve a fracture could be removed.

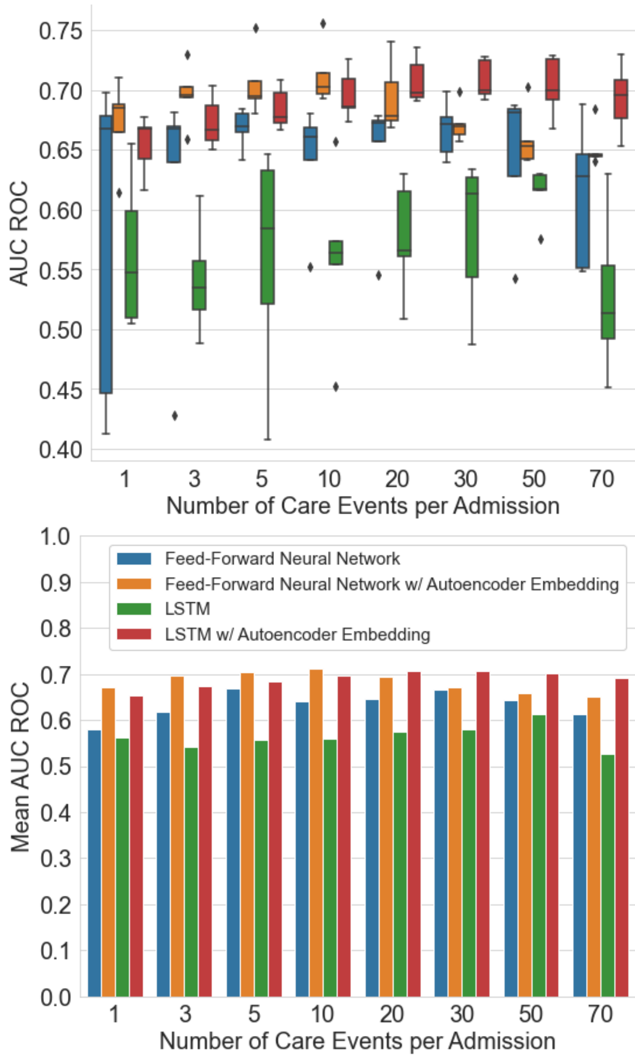

Figure 9. Comparing results of all four models: feed-forward neural network, feed forward neural network with autoencoder embedding, LSTM, and LSTM with autoencoder embeddings. Top graph shows boxplot results of 5-fold cross validation. Bottom graph shows only mean ROC AUC scores across 5-fold cross validation.

5. REFERENCES

[1] MIMIC-III v1.4 documentation. Overview of the MIMIC-III data. https://mimic.physionet.org/gettingstarted/overview/.

[2] Beaulieu-Jones BK, Orzechowski P, Moore JH. Mapping Patient Trajectories using Longitudinal Extraction and Deep Learning in the MIMIC-III Critical Care Database. Pac Symp Biocomput. 2018;23:123-132. PMID: 29218875.

[3] Beaulieu-Jones BK, Greene CS; Pooled Resource Open-Access ALS Clinical Trials Consortium. Semi-supervised learning of the electronic health record for phenotype stratification. J Biomed Inform. 2016 Dec;64:168-178. doi: 10.1016/j.jbi.2016.10.007. Epub 2016 Oct 12. PMID: 27744022.

[4] Miotto, R., Li, L., Kidd, B. *et al.* Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Sci Rep* 6, 26094 (2016). https://doi.org/10.1038/srep26094

[5] Beaulieu-Jones BK, Moore JH. MISSING DATA IMPUTATION IN THE ELECTRONIC HEALTH RECORD USING DEEPLY LEARNED AUTOENCODERS. Pac Symp Biocomput. 2017;22:207-218. doi: 10.1142/9789813207813_0021. PMID: 27896976; PMCID: PMC5144587.

[6] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. Journal of Machine Learning Research, 9:2579–2605, Nov. 2008.

[7] Yu K, Zhang M, Cui T, Hauskrecht M. Monitoring ICU Mortality Risk with A Long Short-Term Memory Recurrent Neural Network. Pac Symp Biocomput. 2020;25:103-114. PMID: 31797590; PMCID: PMC6934094.

[8] MIMIC-III v1.4 documentation. MetaVision. https://mimic.physionet.org/mimicdata/metavision/.

[9] MIMIC-III v1.4 documentation. CareView. https://mimic.physionet.org/mimicdata/metavision/.

[10] Zeiler, Matthew D.. "ADADELTA: An Adaptive Learning Rate Method." *ArXiv* abs/1212.5701 (2012): n. pag.

[11] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. *Neural Comput* 1997; 9 (8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735