NeuralLOS: A neural way to predict length of stay Youtube URL : https://youtu.be/IV4zPRjpNo8

Prashanti Nilayam Rudranil Deb Roy Alan Flanders Bhaskar P

ABSTRACT

Length of stay for inpatient visit plays a critical role in not only patient outcome[10; 25] but also resource planning and management[2] of the hospitals. Inpatient length of stay also impacts readmission rates[6; 17]. Timely prediction of estimated length of stay can have positive impact on mortality rate. It can also help hospital management plan and manage resource utilization efficiently while removing unnecessary hassle of readmission and thus increasing patient satisfaction. It has been researched actively and several traditional machine learning based methods[9; 31] have been proposed for early estimation of the estimated length of stay.

Deep learning methods[22; 15; 30] have shown better performance in healthcare research[26; 12] as compared to traditional machine learning methods. For our research, we propose a deep learning based approach to better predict the estimated length of stay for patients using a CNN[22] model on the commonly collected data during inpatient several visits.

1. RELATED WORK

There has been many important research work done to predict length of stay. David H. Gustafson [8] presented a very early study using Bayesian and regression forecaster which demonstrated that using statistical learning techniques like a Bayesian regression, cost effective prediction of length of stay is possible.

Suresh et al.[32] employed the back propagation concept from neural network to identify the pattern in patient data to predict length of stay. In another study Clark et al. [5] used a Poisson regression to estimate length of stay.

In the last few years, many researchers have explored the possibility of using deep learning to predict length of stay. Gentimis et al. [11] use a simple neural network on MIMIC-III[19] data set and demonstrate that a generic model (not specific to a category of health condition or group of conditions) can be developed and used with high accuracy to predict length of stay.

Rocheteau et al. [29] employ a temporal CNN to capture temporal trends and inter-feature relations to predict length of stay. They train the model on eICU database and evaluate against MIMIC-IV database. The evaluation results presented in the paper shows significant improvement over baseline models like standard LSTM[15], channel-wise LSTM[15] and transformer models.

Clinical notes are one of the most rich data sources captured during patient care. Using clinical notes in model prediction can greatly improve the model accuracy. Huang et al.[16] developed ClinicalBERT based on BERT[7] model to extract insights from notes and apply them to predict hospital readmission.

In another study, Weissman et al. [33] show that including clinical notes in the study can improve prediction of stay and mortality. Mullenbach et al. [24] present us with an approach to extract ICD codes from clinical text.

2. INTRODUCTION

It is quite evident that a lot of work has been previously done to predict length of stay using several statistical machine learning and deep learning techniques. Also, a lot of researchers have started focusing on clinical notes as important source of information on patient health conditions.

While there have been several prior research done to predict length of stay using physiological data collected for the patients, to our knowledge, none of the studies explore the possibility of using both physiological data and the information from the clinical text to predict the remaining length of stay.

We propose a neural architecture NeuralLOS which can learn information from the clinical notes and the physiological data as well. For the first few layers, we will pass both the data sources through separate layers using different architectures to extract the hidden states. We will combine the generated states from the both the sources to create a single data source which will be passed through another set of layers to predict the remaining length of stay.

For the physiological data, we will employ a sliding window approach to capture temporal information of the diagnosis using a CNN model, while for the clinical notes we will generate embeddings using GloVe and pass it through either CNN or RNN architecture.

3. METHOD

3.1 Data

The data for this project comes from the Mimic-III critical care database [19; 27] which contains de-identified health-related data including demographics, vital sign measurements, laboratory test results, procedures, medications, caregiver notes, and more. The data is structured as a relational database and can be used for a wide range of applications including length of stay prediction which is the focus of this

	Train	Test
# Patients	28,620	5,058
# ICU stays	35,621	6,281
Total samples	2,925,434	525,912

Table 1: Benchmark data for length of stay data set

project. Pre-processing of the data makes use of a standard benchmark [12] code but considerable alterations to the code were required to make it performant and to bring in clinical notes which the original benchmark did not include in their models.

Cleansed tabular features and label values are generated by the benchmark to predict risk assessment (mortality prediction), physiologic compensation, phenotype, and length of stay prediction. The benchmark also includes code for generating prediction baseline benchmarks which will be used to compare our results. The statistics for length of stay are shown in Table 1 including the split between training and test datasets.

We append the features generated using above benchmark with clinical notes for the visit. We then convert the notes to embedding which is used as input to the model.

Length of stay prediction labels are created for every period length of the episode (ie. length of stay label decreases as period length increases). CSV files are also created for each patient/episode containing approximately hourly physiological data points including Glasgow comma scale measures, glucose, O2, blood pressure, heart rate and temperature. The data is not currently organized by time and further investigation is required to determine if the benchmark data can be consolidated hourly. This will be required to use our intended approach using a CNN architecture. A CNN requires that we use fixed length periods.

The benchmark dataset does not contain clinical notes and so a clinical note dataset will need to be built as part of the project and grouped into fixed-length time windows. Further investigation is required to determine if this can best be accomplished by altering the benchmark or writing a separate data processing routine. Further investigation will be required to determine the optimal time window to use. Compared to the original paper using clinical notes to predict health outcomes, the MIMIC-III data set generally has shorter stay lengths and so using an 8-hour window may be too long.

3.2 Pre-processing

Mimic-III database has been very popular among researchers for healthcare research. We are using the benchmark [13] to perform initial pre-processing of the raw data from Mimic-III. The benchmark pre-processing generates tabular physiological data with true values of the remaining length of stay along with other patient information for the entire inpatient stay duration. The data is generated as a time-series with all the available observations over the time for each patient for each episode of admission.

The original benchmark code does not process clinical notes. We have modified the benchmark code to collect clinical notes corresponding to the time-series of physiological data. We group these clinical notes into sliding windows and generate the embeddings for the raw notes. These generated embeddings are used as input to our model.

We create similar sliding windows for pre-processed physiological data as well which capture the temporal information. We use 5 hour as the window size for both physiological data and notes embeddings. For example, if a patient has stayed for 7 hours, the generated windows are [0,1,2,3,4], [1,2,3,4,5], [2,3,4,5,6], [3,4,5,6,7]. Each number represents the hour from the time of admission and information is grouped for all features for all the hours in the window.

3.3 Winsorization

During our initial data analysis, we observed that there were a few extreme values in the dataset which did not represent the true characteristics of the data. We applied a winsorization of 94% on the dataset to remove these outliers from the training process. Winsorization is a common statistical technique for handling outliers. A winsorization of 94% removes 3% of extreme data from both ends of the datasets.

3.4 Benchmark Models

From the original benchmark [13] we are comparing against a linear regression and simple LSTM model. The results reported in the original paper developed an ordered classification method instead of using simple regression and predicting the number of days remaining. The custom metric better captures how LOS is used in practice. The range of values is divided into ten buckets, one bucket for extremely short visits (less than one day), seven day-long buckets for each day of the first week, and two "outlier" buckets – one for stays of over one week but less than two, and one for stays of over two weeks. The regression problem of length-of-stay prediction is then converted into an ordinal multi class classification problem. A Kappa score is used to measure this classification because it can be used to measure ordered classes and the correlation between them.

From the original code we trained the standard LSTM model using simple regression for comparison and for better comparison to the linear regression baseline which only trains against the raw LOS value.

Common metrics for regression tasks are Mean Squared Error or Mean Absolute Difference which are use for comparing the models. Mean Squared Error was not used in the original benchmark.

The original benchmark did not take into consideration clinical notes and so this has been added to the benchmark pre-processing.

Although code from the benchmark was used for this project, considerable changes were required to make it compatible with the latest versions of Keras and Tensorflow. Also mutiprocessing support needed to be added in order to make the LOS task performant as it was bottle necked creating the tensors for training.

3.5 Clinical notes

Several studies and research work [16; 7; 18] have shown that clinical notes contain very useful patient information and can be successfully used in a deep learning model. For our research, we evaluated BioClinicalBERT [1] which is a freely available BERT[7] model stemming from BioBERT[23] and has been tuned with MIMIC-III clinical notes, and BioSentVec[4] which uses PubMed[3] to generate embeddings from MIMIC-III clinical notes. Both of these methods generate embeddings of similar shapes. We use these embeddings as input

Layer#	Layer Name	#Input Params	#Output Param
1	Conv2d	5,440	87,040
2	Conv2d	87,040	174,080
2	MaxPool2d	174,080	34,816
3	Conv2d	34,816	69,632
4	Linear	69,632	8,192
5	Dropout	8,192	8,192
6	Linear	8,192	4,096
7	Linear	4,096	1,024

Table 2: PhysioNet: Layerwise Parameters

Layer#	Layer Name	#Input Params	#Output Param
1	Conv2d	1,966,080	7,864,320
2	MaxPool2d	7,864,320	1,966,080
2	Conv2d	1,966,080	3,932,160
3	Conv2d	3,932,160	1,966,080
4	Linear	1,966,080	65,536
5	Dropout	65,536	65,536
6	Linear	65,536	16,384
7	Dropout	16,384	16,384
8	Linear	16,384	4,096
9	Linear	4,096	1,024

Table 3: NotesNet: Layerwise Parameters

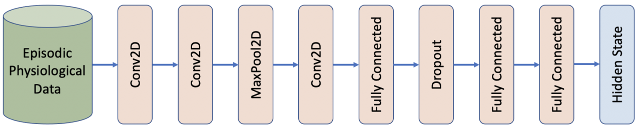

Figure 1: PhysioNet implementation

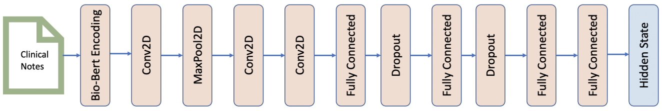

Figure 2: NotesNet implementation

to our NotesNet [fig 2] to learn hidden state from the embeddings.

3.6 NeuralLOS Design

The duration of stay for an inpatient admission depends on several factors. The obvious factors are the physiological data collected from the patient like blood pressure, temperature, etc. However, these features may change over the duration of stay. It is a common practice at a healthcare center to monitor and capture all these features at regular intervals up to the time when patient is either discharged or deceased. This add a temporal component to the dataset in which the remaining length of stay also depends on the change in the patient conditions over time.

In such datasets, opting for a RNN model like LSTM[15], seems like an obvious choice to capture the sequential information in the data. Also, since MIMIC III[27] is a big dataset, processing as much information as possible to train a model seems the reasonable choice. Generally, a neural network model tends to perform better when trained with a bigger dataset. However, training an RNN model on such a large dataset requires considerable amount of hardware resources.

We propose a CNN[22] based neural network architecture NeuralLOS. To capture the temporal information from the dataset, we adopt a sliding window approach. Each window consists of the all the observations from last few hours. These windows are then batched and shuffled to create the training dataset. The remaining length of stay at the end of the window is taken as the true output. For our initial experiments, we have taken a 4-hout window. For example, if an inpatient admission record has observations for 6 hours of stay, the widows are [1,2,3,4], [2,3,4,5] and [3,4,5,6]

For our model design, we explored the possibility of using a pre-trained CNN model like AlexNet[21], ResNet[14], etc. as our base model. However, most of these pre-trained models are trained on images, which might not be ideal for our use case. We went with a simple 9-layer neural network for our implementation.

The architecture of our model is shown in figures [1, 2, 3]. We call our top level model as EpisodeNet. It is a combination of two different models, PhysioNet and NotesNet.

3.6.1 PhysioNet

The architecture of PhysioNet is shown in figure 1. The model consists of 3 convolution layers, one pooling layer, one dropout layer and 3 fully connected linear layers. The model parameters are given in table 2. We use this model to process the tabular physiological data for each batch. The output of the mode is a tensor with (#batch_size, 32) which represents the hidden learnt state of the model from the tabular physiological data.

3.6.2 NotesNet

The architecture of NotesNet is shown in figure 2. The model consists of 3 convolution layers, one pooling layer, 2 dropout layers and 4 fully connected linear layers. We use BioClinicalBERT [1] to generate the embeddings from the notes of the batch. These embeddings are then passed as input to NotesNet to process and learn information for a given batch. The input to this model is a tensor with shape (#batch_size, #sentences, #embeddings). For our model, we have selected #sentences as 80 and #embeddings as 768. The output of the mode is a tensor with (#batch_size, 32) which represents the hidden learnt state of the model from the notes data.

3.6.3 EpisodeNet

EpisodeNet is the top level model which uses both PhysioNet and NotesNet to process each batch of data. The architecture is shown in figure 3. The tabular physiological data from each batch is passed through PhysioNet and the corresponding notes embedding is passed through NotesNet. The output hidden states from both PhysioNet and NotesNet are then concatenated and passed through a couple of fully connected linear layers to predict the remaining length of stay as a regression output.

3.7 Metrics

We selected 3 different metrics commonly used with regression models for our evaluations.

3.7.1 Mean Squared Error

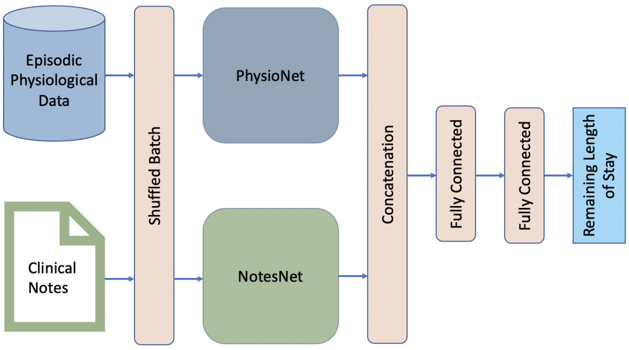

Figure 3: EpisodeNet implementation

MSE measures the average of the square of the errors. It is calculated by taking an average of the square of the difference between the true values and the predicted values. It is sensitive to outliers and has higher penalty with greater deviation on true and predicted values. It is given as:

$$\frac{\sum(y_{true} - y_{pred})^2}{\#sample}$$

3.7.2 Mean Absolute Error

MAE measures the average of the absolute difference of the errors. It is calculated by taking an average of absolute value of the difference between the true values and the predicted values. It is less sensitive to outliers. It is given as:

$$\frac{\sum |(y_{true} - y_{pred})|}{\#sample}$$

3.7.3 Mean Absolute Percentage Error

MAPE measures the average of the ratio of the absolute difference of the errors to the true value. It gives the normalized version of the MAE by true values. It is given as:

$$\frac{100}{\#sample} * \sum |\frac{(y_{true} - y_{pred})}{y_{true}}|$$

3.8 Hyper-parameter tuning and selection

For any deep learning model, hyper-parameter tuning is one of the crucial steps to determine the best parameters for the model. For our model, we experimented various combinations of different hyper-parameters to select the optimal hyper-parameters for best results.

3.8.1 Number of epochs

We experimented with different number of epochs with combination of different hyper-parameters like learning rate, batch size over multiple iterations. We compared the training loss while also calculating the metrics on validation set for each epoch.

We observed that while training loss decreases rapidly and substantially over the first few epochs, it eventually flattens out at around 5 epochs [Figure 4] for all the iterations. Also, while the MSE for validation remains quite similar up to 5 epochs, it begins to increase [Figure 5] after that, which might indicate over-fitting of model on the training set.

Based on our observation we selected the number of epochs as for our evaluation.

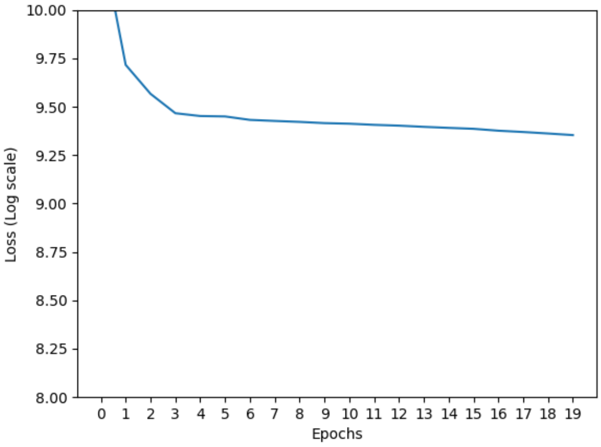

Figure 4: Trend of training losses over epochs

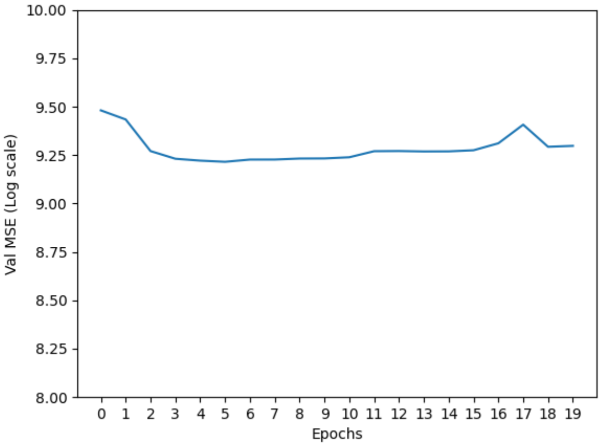

Figure 5: Trend of validation mean squared errors over epochs

3.8.2 Batch size; Learning rate

Similarly, we experimented with different batch sizes with a combination of different hyper-parameters. We selected 32, 128 and 256 as candidate batch sizes and ran multiple iterations with different learning rates. We collected different metrics with all the combinations as shown in tables [4, 5, 6].

Based on these metrics, the model performed best on test dataset when trained with batch size of 32 and learning rate of 0.0001.

3.8.3 Optimizer and Loss function

Since we have designed our model to predict a regression output, we experimented with two different loss functions: 1) L1Loss 2) MSELoss. Based on the test results over multiple iterations, we observed that MSELoss is better suited to our model.

We explored Stochastic gradient descent and Adam[20] optimizers for our model. Based on the experimental results,

lr/batch size	**32**	**128**	**256**
0.01	6.821e+12	1.113e+12	3.555e+11
0.001	9.858e+4	1.009e+5	5.363e+4
0.0001	**1.744e+4**	9.187e+4	7.652e+4
0.00001	1.472e+5	1.427e+5	2.498e+5

Table 4: A comparison of mean square error values on test set for different batch sizes and learning rates

lr/batch size	**32**	**128**	**256**
0.01	3.099e+5	4.809e+4	1.601e+4
0.001	8.239e+1	8.474e+1	8.280e+1
0.0001	**8.070e+1**	8.297e+1	8.327e+1
0.00001	9.132e+1	1.001e+2	9.739e+1

Table 5: A comparison of mean absolute error values on test set for different batch sizes and learning rates

we selected Adam as the optimizer for our model.

4. RESULTS

4.1 Evaluation

The initial results of the benchmark models are close to the benchmark results [12] with our results being slightly better. For example, in the original paper a basic LSTM model resulted in an MAE of 94.7. Our score was 79.3. We performed the training using a regression output instead of using custom bins (classification problem). Our results when training against custom bins (ie, 1 day, 2 days, 3 days, ..., 1 week, 2 weeks) did not produce good results.

See Table 7 for a full listing of results.

When evaluating a forecasting model it's useful to know a) both how much historical data is required to make good predictions and b) how close to the last state does the model make good predictions. [28].

In order to test the model at different intervals we first applied Winsorization of 96% across length of stay for an episode. Those in the bottom 3% and top 97% were removed. We also then removed any episode less than 60 hours so that each could be compared consistently at different time periods. 60 hours was chosen because was close to 1 standard deviation of the average length of stay (66). Figure 6 shows the distribution of episodes over length of stay after Winsorization was applied.

There are 1,555 episodes in the test set to consider. Figure 7 shows that the number of episodes at each period up to 60 have a consistent number of data points. If we plot mean squared error at these different time periods we see how the models perform as they have access to more information.

lr/batch size	**32**	**128**	**256**
0.01	1.008e+4	1.528e+3	3.337e+3
0.001	**1.636**	1.845	1.765
0.0001	1.662	1.706	1.737
0.00001	2.349	2.839	2.596

Table 6: A comparison of mean absolute percentage error values on test set for different batch sizes and learning rates

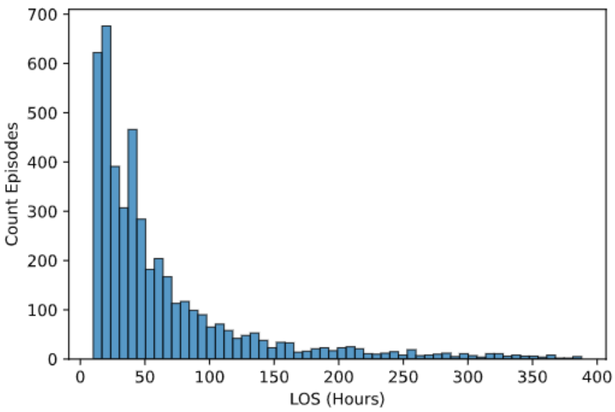

Figure 6: Histogram showing episodes over length of stay after Winsorization

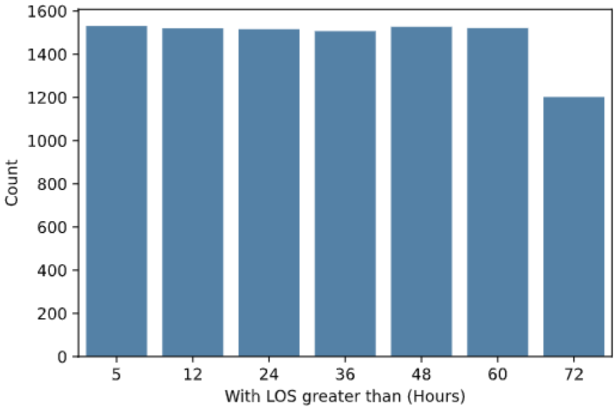

Figure 7: Number of data points at different lengths of stay

The results are shown in figure 8.

As expected the error decreases at progressive time periods for all models except for linear regression which comes down up to 50 hours, but then goes back up. Other models also seem to have an inflection point at 60 hours except for NeuralLOS with full data and LSTM.

When comparing NeuralLOS with only physiological (tabular) data and a model augmented with Bio-ClinicalNote embeddings [16] we included a version of NeuralLOS that was trained on the same data set as the model with notes. Processing notes takes a significant amount of time and we were not able to train on the full training data set. In order to create a comparison we trained NeuralLOS on the same smaller data set in order to try and see differences. The model with notes appears to perform better than the tabular-only model overall, but not when we only look at stays longer than 60 days.

We also looked at whether the predictions become more accurate the closer the patient gets to their end of stay. Using the same episodes we created bins from 2-weeks (336 hours) to 12 hours. The number of data points in each bin is shown in figure 9.

The results again are as expected in that all of the models get better as they approach the final stay. Surprisingly linear regression outperforms the other models earlier on and then

Model	Data types	MAE	MSE	MAPE
Linear regression	Tabular	121.69	12,805,595	3.15
LSTM	Tabular	**79.28**	16,889	81.10
LSTM	Notes	101.44	28,201	**0.72**
PhysioNet (full data)	Tabular	78.55	17,492	1.01
PhysioNet (Part data)	Tabular	80.50	17,450	1.66
PhysioNet+Notes (Part data)	Tabular+Notes	80.49	**16,122**	1.55

Table 7: Model results for length of stay prediction.

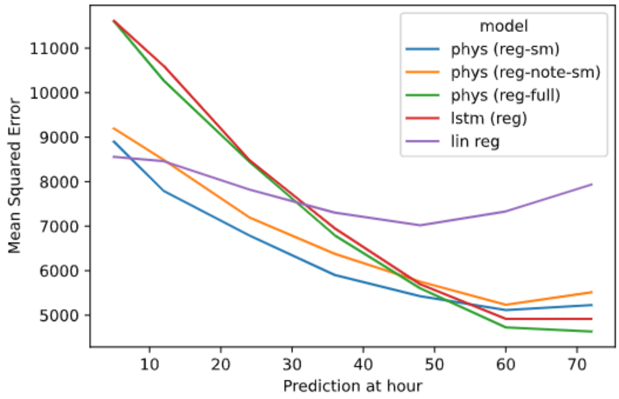

Figure 8: Mean Squared Error at different hours of stay

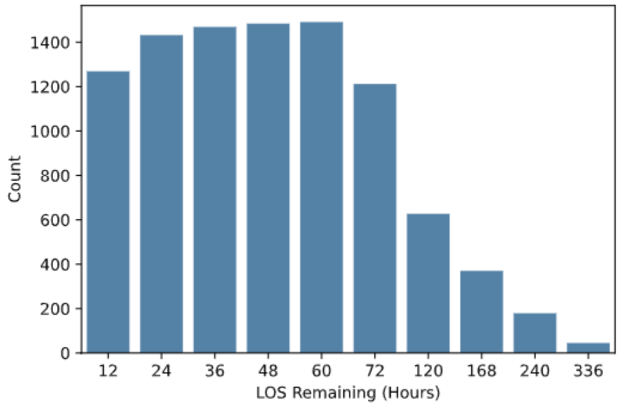

Figure 9: Number of data points at different remaining length of stay

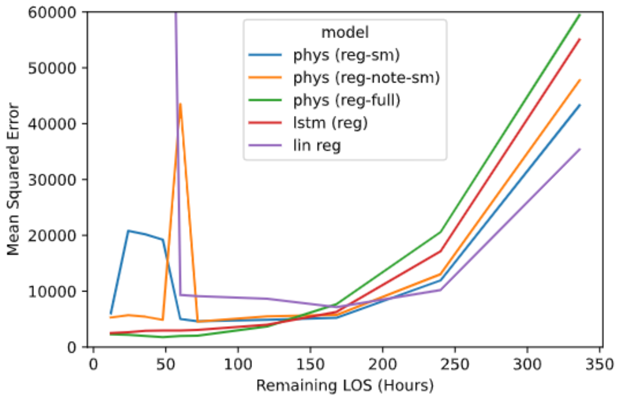

Figure 10: Mean Squared Error at different remaining lengths of stay

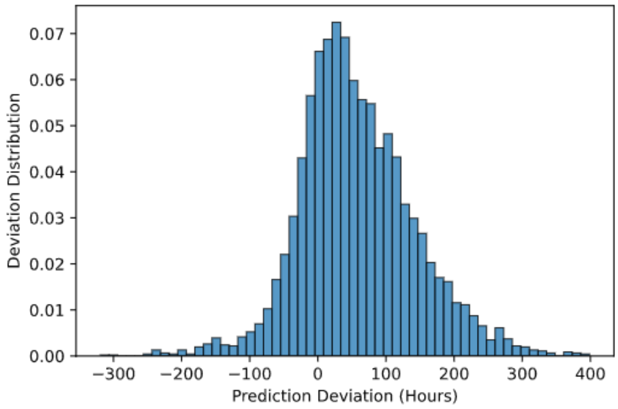

Figure 11: Linear regression deviation distribution

has an inflection point at 168 hours and then goes off the chart at 60 hours. The NeuralLOS and LSTM models start out worse but become better at about 120 hours.

Plotting MSE for each model allows us to compare the models, but it can be difficult to see exactly how a model is performing. By plotting a histogram of deviation in hours we can visualize the spread and degree of the model accuracy. Figures 11 through 14 show this distribution across the models.

The accuracy at each remaining length of stay can be see using a series of box plots which are shown in figures 15 through 18.

4.2 Infrastructure

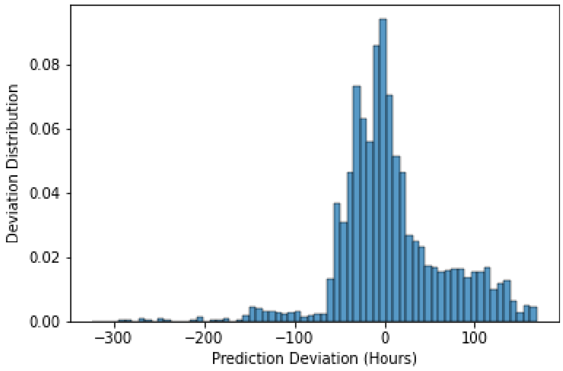

Figure 12: LSTM deviation distribution

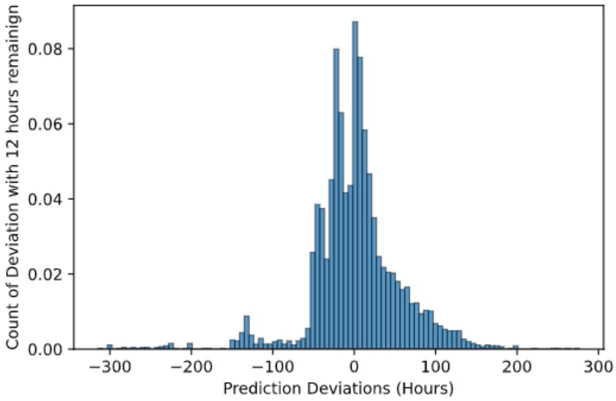

Figure 13: NeuralLOS deviation distribution

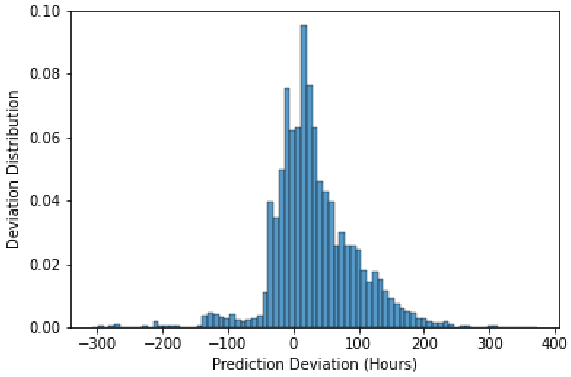

Figure 14: NeuralLOS with notes deviation distribution

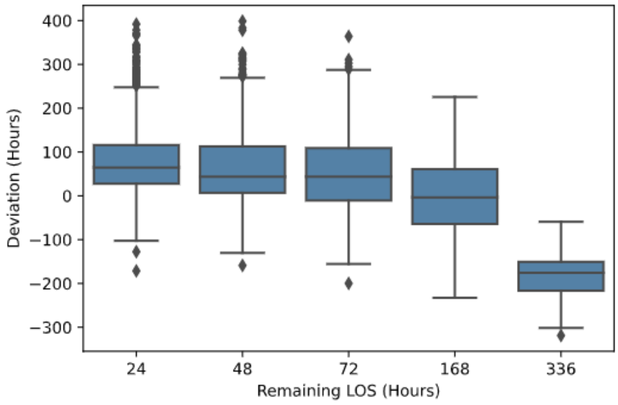

Figure 15: NeuralLOS with notes deviation distribution

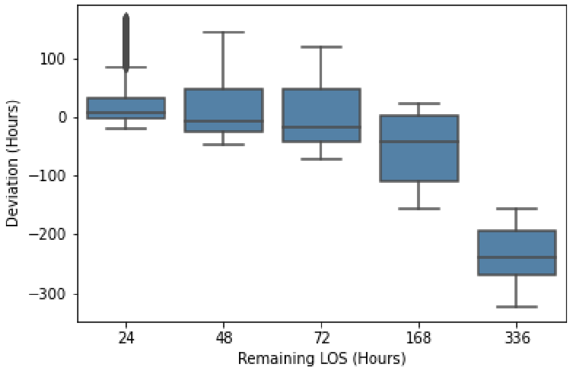

Figure 16: LSTM deviation distribution

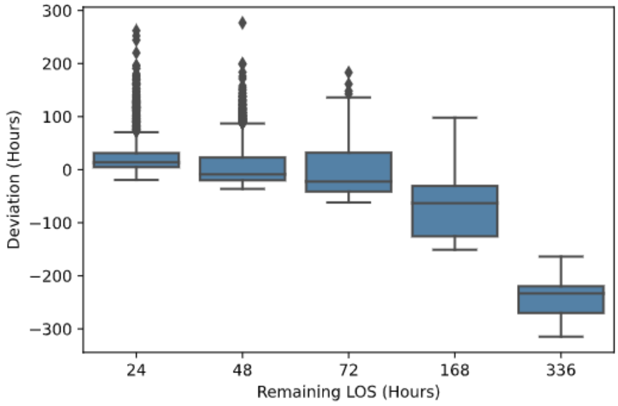

Figure 17: NeuralLOS deviation distribution

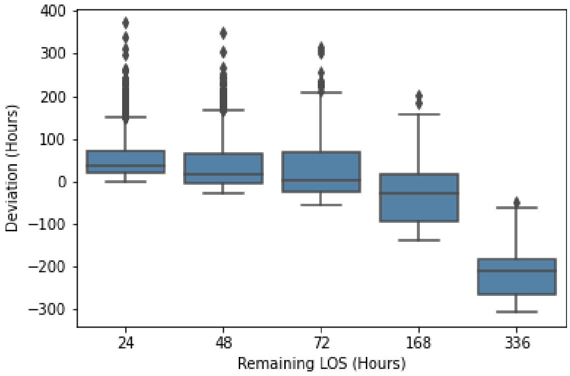

Figure 18: NeuralLOS with notes deviation distribution

We trained and evaluated our models on Google Cloud Platform. The machine configuration is listed below:

Machine Type	n1-standard [16 vCPUs]
CPU Platform	Intel Broadwell
Memory	110GB
GPU	NVIDIA Tesla 4
Storage	400 GB SSD

We are using popular python libraries including but not limited to pytorch, keras, tensorflow, scikit-learn, matplotlib, pickle. The code is available in a GitHub repository. Note that the benchmark code is constrained by data pre-processing intensive tasks. The benchmark code did not have the capability to run in parallel in utilize GPU resources effectively. We invested considerable amount of effort to make the data pre-processing multi-threaded so as to make the best use of the GPU.

5. TEAM CONTRIBUTIONS

It was a team work, as all the members worked in almost in all areas either contributing towards the plan or working or experimenting on data that helped in training and building the model.

Alan set up the GCP environments for training the LSTM and linear regression models. The benchmark code we used required a lot of massaging and upgrading to work correctly. Newer versions of libraries including TensorFlow and Keras required changes to the code. There were also challenges wrt speed which required implementing multi-processing capabilities to the pre-processing routines used to create the tensors for training. Alan wrote the Data and Evaluation sections of the report and wrote the program that combines the results from all models.

Prashanti worked on the NeuralLOS model architecture architecture and the windowing of the dataset. He participated in training the model and metrics generation.

Bhaskar and Rudra participated in the generation of BioSentVec and BioClinicalBert to generate the notes embedding. They also participated in generating the pre-proccesed data from benchmark code.

6. CONCLUSIONS

Predicting length of stay is an important healthcare problem. Getting an estimate of remaining length of stay can help the hospital have a better planning of healthcare resources. It can also be useful for insurance companies for estimating expenses. Using NeuralLOS, we were able to have very good results in predicting length of stay. We compared our model to various benchmark models and presented our results with different perspectives. While, some more work is needed to improve the model further, NeuralLOS, even in the current implementation is producing better results.

7. LIMITATIONS

One of the major limitations we faced, was computation resources to process the entire dataset. The notes embeddings take up a lot of memory and due to memory limitations we could not fit the entire working set in memory. Also, since NeuralLOS computes a lot of parameters, we needed GPUs to make it train faster. After some challenges, we got the one GPU in GCP with limited capacity. Due to these limitations, we trained our EpisodeNet on a subset of data.

We observed that the prediction results from NeuralLOS gets better for pateints with longer stays. This can be attributed to the fact that with time, more information is accumulated and the model is able to predict better.

8. RESOURCES

Youtube URL : https://youtu.be/IV4zPRjpNo8
Github : https://github.com/PNilayam/CS598_DLH

9. ACKNOWLEDGEMENTS

This work was supported by Professor Jimeng Sun and all TAs We would like to thank them for their guidance.

10. REFERENCES

[1] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. B. A. McDermott. Publicly available clinical bert embeddings, 2019.

[2] H. Baek, M. Cho, S. Kim, H. Hwang, M. Song, and S. Yoo. Analysis of length of hospital stay using electronic health records: A statistical and data mining approach. *PloS one*, 13(4):e0195901, 2018.

[3] K. Canese and S. Weis. Pubmed: the bibliographic database. In *The NCBI Handbook [Internet]. 2nd edition.* National Center for Biotechnology Information (US), 2013.

[4] Q. Chen, Y. Peng, and Z. Lu. Biosentvec: creating sentence embeddings for biomedical texts. *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, Jun 2019.

[5] D. E. Clark and L. M. Ryan. Concurrent prediction of hospital mortality and length of stay from risk factors on admission. *Health services research*, 37(3):631–645, 2002.

[6] S. Cropley. The relationship-based care model: evaluation of the impact on patient satisfaction, length of stay, and readmission rates. *JONA: The Journal of Nursing Administration*, 42(6):333–339, 2012.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] G. DH. Length of stay: Prediction and explanation. *Health services research, 3(1), 12–34.*, 1968.

[9] J. Fang, J. Zhu, and X. Zhang. Prediction of length of stay on the intensive care unit based on bayesian neural network. In *Journal of Physics: Conference Series*, volume 1631, page 012089. IOP Publishing, 2020.

[10] R. Figueroa, J. Harman, and J. Engberg. Use of claims data to examine the impact of length of inpatient psychiatric stay on readmission rate. *Psychiatric Services*, 55(5):560–565, 2004.

[11] T. Gentimis, A. J. Alnaser, A. Durante, K. Cook, and R. Steele. Predicting hospital length of stay using neural networks on mimic iii data. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 1194–1201, 2017.

[12] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1), Jun 2019.

[13] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] K. Huang, J. Altosaar, and R. Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

[17] S.-J. Jang, I. Yeo, D. N. Feldman, J. W. Cheung, R. M. Minutello, H. S. Singh, G. Bergman, S. C. Wong, and L. K. Kim. Associations between hospital length of stay, 30-day readmission, and costs in st-segment–elevation myocardial infarction after primary percutaneous coronary intervention: a nationwide readmissions database analysis. *Journal of the American Heart Association*, 9(11):e015503, 2020.

[18] B. L. S. G. C. P.-P. J. M. A. V. M. M. Jienan Yao, Yuyang Liu and M. Ghassemi. Visualization of deep models on nursing notes and physiological data for predicting health outcomes through temporal sliding windows. In *Explainable AI in Healthcare and Medicine*, pages 115–129, 2021.

[19] A. E. Johnson, T. J. Pollard, L. Shen, L. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[22] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[23] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, Sep 2019.

[24] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018.

[25] K. J. Ottenbacher, P. M. Smith, S. B. Illig, R. T. Linn, G. V. Ostir, and C. V. Granger. Trends in length of stay, living setting, functional outcome, and mortality following medical rehabilitation. *Jama*, 292(14):1687–1695, 2004.

[26] A. Peimankar and S. Puthusserypady. Dens-ecg: A deep learning approach for ecg signal delineation. *Expert Systems with Applications*, 165:113911, 2021.

[27] A. E. Pollard, Tom J abd Johnson. The mimic-iii clinical database. http://dx.doi.org/10.13026/C2XW26, 2016.

[28] A. e. a. Rajkomar. Scalable and accurate deep learning with electronic health records. 6:18, 2018.

[29] E. Rocheteau, P. Liò, and S. Hyland. Temporal pointwise convolutional networks for length of stay prediction in the intensive care unit. *arXiv preprint arXiv:2007.09483*, 2020.

[30] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[31] M. Sotoodeh and J. C. Ho. Improving length of stay prediction using a hidden markov model. *AMIA Summits on Translational Science Proceedings*, 2019:425, 2019.

[32] A. Suresh, K. Harish, and N. Radhika. Particle swarm optimization over back propagation neural network for length of stay prediction. *Procedia Computer Science*, 46:268–275, 2015. Proceedings of the International Conference on Information and Communication Technologies, ICICT 2014, 3-5 December 2014 at Bolgatty Palace Island Resort, Kochi, India.

[33] G. E. Weissman, R. A. Hubbard, L. H. Ungar, M. O. Harhay, C. S. Greene, B. E. Himes, and S. D. Halpern. Inclusion of unstructured clinical text improves early prediction of death or prolonged icu stay. *Critical care medicine*, 46(7):1125, 2018.