

Hospital Readmission Prediction Of ICU Patients Using Deep Learning Algorithms

by

Shivam Sinha

Roll. No.: 2014IPG-082



विश्वजीवनामृतं ज्ञानम्

**ABV-INDIAN INSTITUTE OF INFORMATION
TECHNOLOGY AND MANAGEMENT GWALIOR (M.P.),
INDIA**

- For Medicare patients, hospitalizations can be stressful, even more so when they result in subsequent readmissions. A number of studies show that hospitals can engage in several activities to lower their rate of readmissions, such as clarifying patient discharge instructions, coordinating with post-acute care providers etc.
- The purpose of this thesis is to use deep neural networks which can work effectively to predict the result that can be utilized to avoid unnecessary hospital readmissions.

The primary objective of this thesis is:

- To develop Deep Neural Network(DNN) models for Healthcare application i.e Hospital Readmission.
- To predict hospital readmission of patients using deep neural networks.
- To compare results between Deep Learning Models and with exiting algorithms.

Novelty of the proposal

- With this thesis, the aim is to use deep learning algorithms to overcome the drawbacks of conventional machine learning algorithms.
- The thesis aims to use nature inspired algorithms for feature extraction.
- The thesis also aims to use hybrid model (Convolution Recurrent Neural Network) which can further increase the accuracy of the proposed model.

Methodology

Flow Diagram

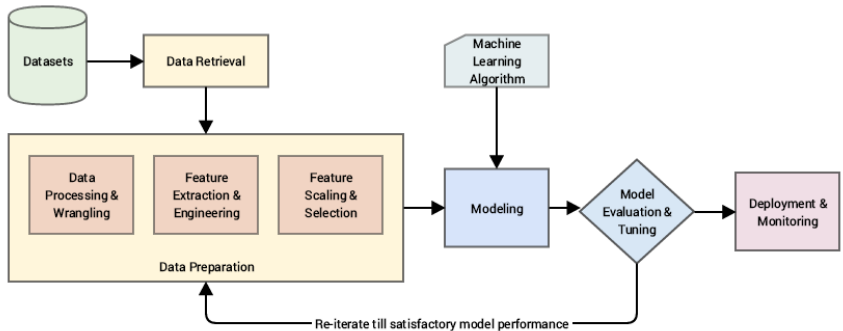


Figure: Flow Diagram

Medical Information Mart for Intensive Care (MIMIC-III)[4] consists of data about patients admitted to various critical care units in a large hospital. A large number of different parameters are present in MIMIC III database. These parameters include information such as vital signs, medications, laboratory measurements, observations, fluid balance, procedure codes, diagnostic codes, imaging reports, hospital length of stay, survival data, and others. The database consists of information of around 58,576 distinct patients who were admitted to various critical care units of the hospital between 2001 and 2012. The data comprises of patients aged 16 years or above only.

Methodology Contd..

Dataset Preprocessing

- Since the dataset is very large, we only consider data of those patients who were readmitted again which gives the details of 7,534 patients. The data set is divided into two classes:-
 - Patients readmitted in 30 days.
 - Patients readmitted after 30 days.
- MIMIC-III dataset contains missing values in some of the features.
 - If feature contains large number of missing values then the feature is removed.
 - If feature contains fewer missing values than mean is used to fill the missing values.

Methodology Contd..

Dataset Preprocessing

- Normalization is used to remove the biases among the features. It brings the data on a standard scale. It standardizes the range of independent features or variables of data, called feature scaling. Min Max Normalization technique is used.

$$v' = (v - \min_A) / (\max_A - \min_A) \quad (1)$$

where A can be vector.

Features are selected through Nature Inspired Algorithms:

- **Bat Algorithm** is an optimization algorithm inspired by the echolocation behavior of microbats. Bats use echolocation to sense distance, and they also distinguish between food/prey and other background barriers.
- **Grey Wolf Optimizer (GWO)** Algorithm inspired by grey wolves. The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves are employed for simulating the leadership hierarchy such as alpha, beta, delta, and omega. Also, the three main steps of hunting, searching for prey, encircling prey, and attacking victim, are implemented.

Methodology Contd..

Feature Extraction

Features are extracted through Convolution Neural Networks:

Convolutional Neural Network is an effective machine learning technique from the deeplearning and it is similar to ordinary Neural Networks. Convolutional neural network is a network with convolutional layers. Convolutional neural network is consists of three steps of neural layers to build its architectures: Convolutional, Pooling, and Fully-Connected.

- **Recurrent Neural Network (RNN)** are used to capture the temporal dependency in the time series data. We are usually provided with a series of observations $x_1 \dots x_T$ and we train a classifier to generate hypotheses \hat{y} .
- **Long Short Term Memory** networks usually just called LSTMs are a special kind of RNN, capable of learning long-term dependencies. Remembering information for long periods of time is practically their default behavior.
- **Bidirectional Recurrent Neural Networks** also called BRNN is just like RNN but it trains simultaneously on both sides of the time series data. This model gives the better result in both regression and classification problem.

- **Auto Encoders** are an unsupervised learning method, although technically, they are trained using supervised learning methods, referred to as self-supervised. They are typically trained as part of a broader model that attempts to recreate the input.

For a given dataset of sequences, an encoder-decoder LSTM is configured to read the input sequence, encode it, decode it, and recreate it. The performance of the model is evaluated based on the model's ability to recreate the input sequence.

Results

Model Implementation Details

- The entire dataset was split using stratified sampling and 10% of the dataset was used as a validation set.
- The rest 80% of the dataset was used for training purpose and was tested on 10% of the dataset.
- The LSTM model was trained with 30 epochs using Adam optimizer. LSTM layer uses 100 memory cell with no dropout and 25 memory cell with 20% dropout and these architectures are found after validation performance. Also train this model with Bat Algorithm.
- CLSTM model was trained with 30 epochs and the batch size of 1000. First, the convolution layer was used with 45 filters of size 5×5 . Then the LSTM layer that uses 100 memory cell with 20% dropout. Also train this model with Grey Wolf Optimizer Algorithm.

Results

Comparison among different deep learning models

Methods	Area Under ROC Curve	Accuracy
Long Short Term Memory	0.88	88.71%
Long Short Term Memory Auto-Encoders	0.73	75.00%
Convolution Long Short Term Memory	0.87	85.63%
Long Short Term Memory with Bat Algorithm	0.75	77.58%
Long Short Term Memory with Grey Wolf optimizer Algorithm	0.85	87.48%

Table: Comparison among different Models implemented so far

Results

Comparison among different deep learning models

Methods	Area Under ROC Curve	Accuracy
Convolution Long Short Term Memory Auto-Encoders	0.87	82.41%
Bi-Directional Long Short Term Memory	0.85	83.42%

Table: Comparison among different Deep Learning Models

Results

Comparison with Baseline Models

Methods	Area Under ROC Curve	Accuracy
Logistic Regression	0.54	58.09%
Support Vector Machines	0.54	59.37%
Decision Tree	0.52	55.68%
Random Forest	0.53	60.49%

Table: Comparison with Baseline Models

Results

Receiver Operating Characteristic Curves

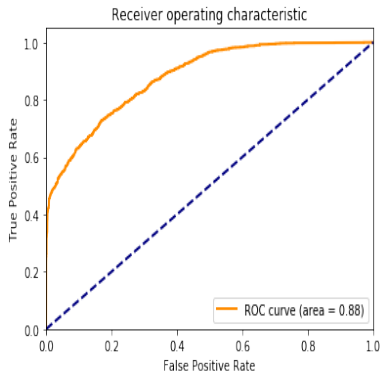


Figure: Receiver Operating Characteristic Curve Long Short Term Memory.

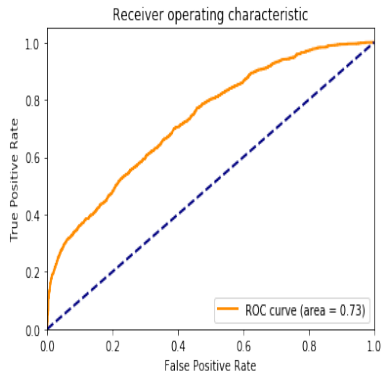


Figure: Receiver Operating Characteristic Curve Long Short Term Memory Auto Encoders.

Results

Receiver Operating Characteristic Curves

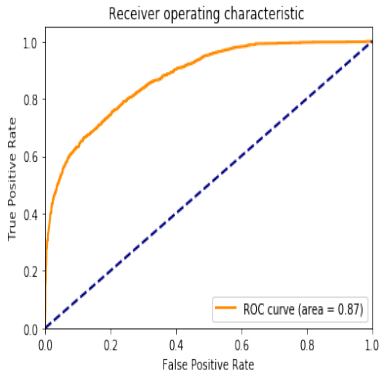


Figure: Receiver Operating Characteristic Curve Convolution Long Short Term Memory.

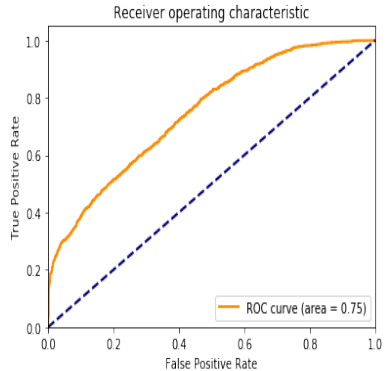


Figure: Receiver Operating Characteristic Curve Long Short Term Memory with Bat Algorithm.

Results

Receiver Operating Characteristic Curves

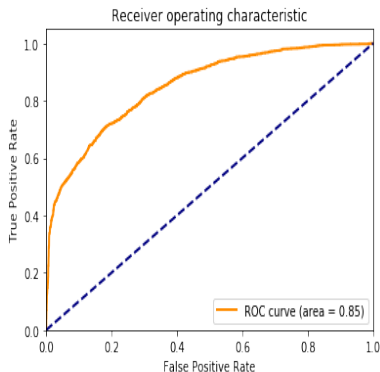


Figure: Receiver Operating Characteristic Curve Long Short Term Memory with Grey Wolf Optimizer Algorithm.

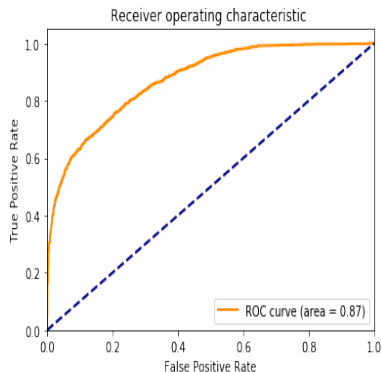


Figure: Receiver Operating Characteristic Curve Convolution Long Short Term Memory Auto Encoders.

Results

Receiver Operating Characteristic Curves

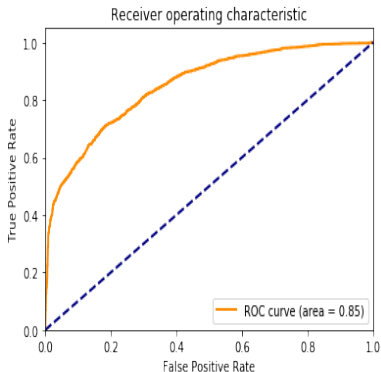


Figure: Receiver Operating Characteristic Curve Bi-Directional Long Short Term Memory.

Results

Model Loss Curves

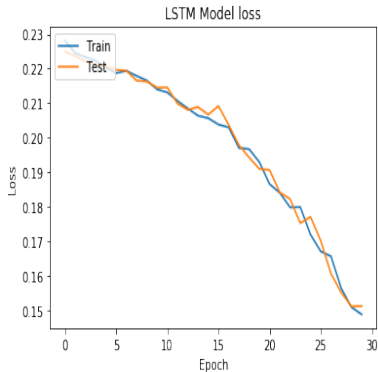


Figure: Loss Curve:LSTM.

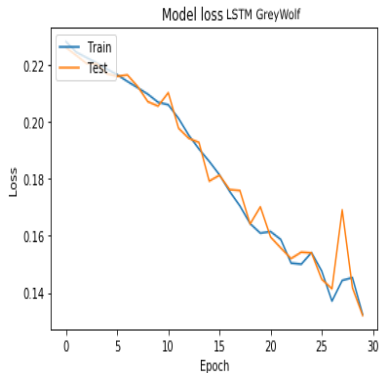


Figure: Loss Curve: LSTM Grey Wolf.

Results

Model Loss Curves

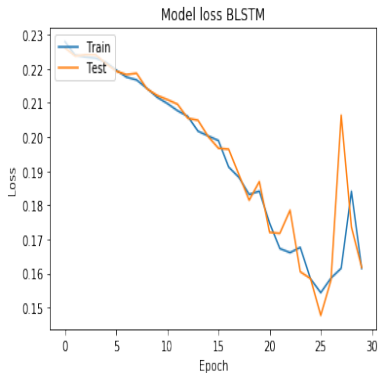


Figure: Loss Curve: Bidirectional LSTM.

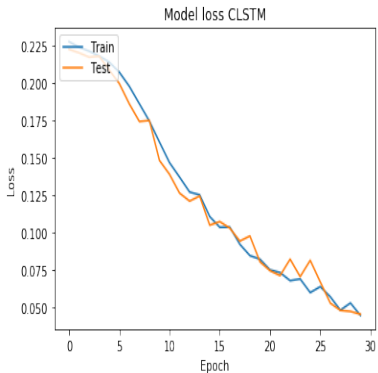


Figure: Loss Curve: Convolution LSTM.



Mohammad M Ghassemi, Stefan E Richter, Ifeoma M Eche, Tszyi W Chen, John Danziger, and Leo A Celi.

A data-driven approach to optimized medication dosing: a focus on heparin.

Intensive care medicine, 40(9):1332–1339, 2014.



Romain Pirracchio, Maya L Petersen, Marco Carone, Matthieu Resche Rigon, Sylvie Chevret, and Mark J van der Laan.

Mortality prediction in intensive care units with the super icu learner algorithm (sricula): a population-based study.

The Lancet Respiratory Medicine, 3(1):42–52, 2015.

References II



Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu.

Interpretable deep models for icu outcome prediction.

In *AMIA Annual Symposium Proceedings*, volume 2016, page 371. American Medical Informatics Association, 2016.



Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.



Bhargava K Reddy and Dursun Delen.

Predicting hospital readmission for lupus patients: An rnn-lstm-based deep-learning methodology.

Computers in biology and medicine, 101:199–209, 2018.



Jonathan Franco, Francesc Formiga, Joan-Carles Trullas, P Salamanca Bautista, Alicia Conde, Luis Manzano, Raúl Quirós, Álvaro González Franco, Alejandro Martín Ezquerro, Manuel Montero-Pérez-Barquero, et al.

Impact of prealbumin on mortality and hospital readmission in patients with acute heart failure.




European journal of internal medicine, 43:36–41, 2017.



Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber.

Lstm: A search space odyssey.

IEEE transactions on neural networks and learning systems, 28(10):2222–2232, 2017.

-  Jack Hanson, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou.
Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks.
Bioinformatics, 33(5):685–692, 2016.
-  Ming Liang and Xiaolin Hu.
Recurrent convolutional neural network for object recognition.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375, 2015.
-  Xin-She Yang and Amir Hossein Gandomi.
Bat algorithm: a novel approach for global engineering optimization.
Engineering Computations, 29(5):464–483, 2012.



Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis.

Grey wolf optimizer.

Advances in engineering software, 69:46–61, 2014.