Hospital Readmission Prediction of ICU Patients Using Deep Learning Algorithms

Saumil Maheshwari
Department of Information Technology)
ABV-Indian Institute of Information Technology and Management
Gwalior, India
saumlimaheshwari@yahoo.co.in

Shivam Sinha
Department of Information Technology)
ABV-Indian Institute of Information Technology and Management
Gwalior, India
ipg_2014082@iiitm.ac.in

Dr. Ritu Tiwari
Department of Information Technology)
ABV-Indian Institute of Information Technology and Management
Gwalior, India
tiwariritu2@gmail.com

***Abstract*—Deep learning healthcare applications have evolved over the last few years. This advancement leads to new applications and possibilities in the field of health care. Due to different variants of deep learning algorithms like convolutional and recurrent neural network these advancements in healthcare application made possible.**

***Index Terms*—Deep learning, LSTM, Healthcare, Recurrent Neural Network, GRU.**

I. Introduction

A. Hospital Readmission

Hospital Readmissions are distinguished as a sign of the low quality of consideration, for example, inadequate release arranging and care coordination. Emergency clinic readmission[1] happens when a patient release from a medical clinic is readmitted inside a predetermined interim. Admittance of a patient can happen due to any decided or undecided reasons. These factors drove Health care associations to consider managing readmission rates; an expansion in readmission rates prompts an increment in punishments. There has been an expanded use of Readmission rates as a result rule in wellbeing administrations investigation and as a standard criterion for well being frameworks. For Medicaid inmate, hospitalizations are exceptionally distressing, and it is considerably more when they occur in frequent re-admissions. A ton of research concentrates demonstrated that emergency clinics could take part in a few exercises like explaining patient release directions, planning with post-intense consideration suppliers, lively cleaning system to lessen the rate of readmissions of patients. Be that as it may, these individual subsequent meet-ups can be costly.

B. Machine Learning For Health Care

Machine learning leads to provide intelligence by providing new knowledge. Since the onset of machine learning, it has been extensively utilized in achieving paramount goals as the healthcare sector is evolving. The healthcare sector is producing an enormous mass of data, and it is beyond human capability to manually analyze this vast data and produce inferences based on that. Machine learning has proven to be a benefit at this level. The machine learning algorithms automatically find patterns in the data, which enable to get inferences from the data easily. This can not only lower the lifetime risk of the patient but also can save millions of dollars if the algorithm can predict efficiently and correctly. Other applications of machine learning in the healthcare domain includes drugs combination which should be avoided taking together, classifying images for different diseases like skin cancer. Keeping this in the picture, it can be said that machine learning can revolutionize the healthcare sector.

C. Deep Learning

Deep learning is one of the extended section of Machine learning(ML), which works on a specific ML algorithm, that is the artificial neural network. It's leveraging the recent advancements in computing power and thus using specific purpose neural network to learn from a plethora of data and make predictions based on the patterns detected. The structure consists of several hidden layers which are meant to extract the information by exploiting the structures present in the data. Deep neural networks are specialized in solving problems, including data, which is greatly structured. It also pledge to substitute hand-engineered attributes with dimensionality reduction methods that are hierarchical, unsupervised, or semi-supervised. Multiple hidden units are being used between input and output in case of Deep Neural Network (DNN). Similar to the artificial neural network, the complex non-linear correlation can be modeled using DNN[2].

D. Recurrent Neural Network

An artificial neural network has a specific kind called recurrent neural network(RNN), and it has directed cycles between connection. The internal state of the network exhibits dynamic temporal behavior. Unlike ANN, random series can be managed by using the internal state of RNN. This is the fundamental structure introduced in the 1980s: neuron-like units in a network, a directed connection between every neuron. There is a time-dependent continuous valued activation in every unit. There are input, output, and hidden layers. One input vector is supplied at every time steps. At every time step, the non-linear function of the weighted sum of all the activations of connected units is calculated by each non-input unit[3].

E. Long Short term Memory

(LSTM) network, a deep learning RNN, is used by numerous researchers, published by Schmidhuber & Hochreiter in 1997[4]. This deep learning system fix the the vanishing gradient issue of standard RNN. The recurrent gates called forget gates are introduced in LSTM. There are an unlimited number of virtual layers in LSTM-RNNs, through which the error can flow backward. LSTM can learn from events that occurred thousands of time steps ago so it can be used for very deep learning tasks. LSTM has a unique architecture that consists of a memory cell, which can keep the state which it is currently in, over duration. Non- Linear gating units are used in LSTM to control the course of data in and out of the memory cell. Over time, many improvements have been suggested and made in the standard architecture of LSTM to make it more efficient. Today, LSTMs are being used in a variety of learning problems which differ in scale and nature significantly when compared to the problems which were initially solved by LSTM.

II. Literature Review

Over the past decade, several studies are accomplished in the area of healthcare. Hospital readmission prediction is vital and essential part of healthcare for the better excellence of life of a man or woman.

After impressive accomplishment in cognitive domains, deep learning is anticipated to reshape healthcare. Most amazing outcomes to date in health have been in diagnostic imaging, that is a natural step given record-breaking outcomes in computer vision. However, diagnostic imaging is simplest a small part of the tale. A full wise medical design has to be capable of cause approximately the past (historic sickness), present (prognosis) and future (analysis). Here we acquire the idea of reasoning as "scientifically utilizing historically gained accomplishments in order to solve new problems". For that, we figure out how to install discrete medical items into continuous vectors, which loan themselves to a wide host of amazing arithmetical and factual administrators.

Author [5] developed an approach that helps clinicians determine the ideal beginning portion of medication to securely and rapidly achieve a helpful aPTT window. In [6] author uses a super learner algorithm is used for predicting hospital mortality in patients. In [7] author used a novel called Interpretable Mimic Learning is utilized, to learn interpretable phenotype attributes for making powerful forecast while imitating the presentation of deep learning models metaheuristics. In [8] Combination of convolution and recurrent neural network model is implemented for object recognition. In [9] authors objective was to reduce overfitting using batch normalization. In [10] author introduced interpretable mimic learning that uses gradient boosting trees to learn interpretable models and in the mean time attains powerful forecast achievements as deep learning models. In [11] author found the proportional split of data is useful for deep learning applications when managing huge volumes of information. In [12] author leveraged longitudinal EHR information for early observation of heart problems using Recurrent Neural Network. Specialists not long ago started endeavouring to apply neural network based strategies (or deep learning) to EHR to use its capacity to learn complex examples from the information. Past examinations, for example, phenotype learning or representation learning , in any case, have not entirely tended to the sequential nature of EHR is particularly identified with our work in that the two investigations use RNN for grouping forecast. Be that as it may, while uses regular times series of real-valued factors gathered from ICU patients to anticipate diagnosis codes, we utilize discrete therapeutic codes (for example diagnosis, medicine, strategy) extricated from longitudinal patient visit documents. Likewise, in each visit, we make a forecast about foresee conclusion, prescription request in the following visit and the opportunity to follow visit. In [13] author uses Long short term memory (LSTM) RNN to predict disordered proteins, such as SPOT-disorder. In [14] State-of-the-art deep and recurrent neural network models were applied to achieve robust results on classifying opioid users. In [15] author used Gated Recurrent Unit (GRU) to fx the vanishing gradient problem of a standard RNN. In [16] author used LSTM to overcomes the vanishing gradient issue of traditional RNN. In [17] author merge the use of time frequency heat map presentations with a deep convolutional neural network (CNN) to classify heart sounds. In [18] author used Statistical Analysis is used to find the mortality and readmission rates. In [19] author implemented a 13-layer deep learning CNN model for the self-regulating EEG investigation.

III. Methodology

A. Data description: MIMIC-III

Medical Information Mart for Intensive Care (MIMIC-III)[20] contains of information about patients enrolled in different critical care units in a large health care center. MIMIC-III is generally viewed as a large and single-center database. A large number of different parameters are present in MIMIC III database. These parameters include information for example vital signs, medicines, laboratory estimations, observations, fluid balance, procedure codes, diagnostic codes, imaging reports, medical clinic length of stay, survival information, and others. The database consists of information of around 58,576 distinct patients who were enrolled to various critical care units of the hospital between 2001 and 2012. The data comprises of patients aged 16 years or above only. Descriptive statistics were performed on this dataset, and it was found that the median age is 65.8 years for adult patients(Q1-Q3: 52:877:8). Out of total patients, there were 55.9% male patients and only 11.5% of the cases had in-hospital mortality. It was calculated from the data that on an average a patient stayed for 2.1 days in ICU and had an average of 6.9 Days of hospital stay. MIMIC-III database have several idiosyncratic properties about it, and these are mentioned below:

- The dataset is a huge database accumulated throughout more than a decade and consists of very detailed information of each patient under care.

- MIMIC-III database requires a user to oblige by a user data agreement and once itis accepted the analysis is boundless.
- It contains discharge summaries as well as reports of ECG, imaging studies and information about various codes like International Classification of Disease, 9thEdition (ICD-9) codes, Diagnosis Related Group (DRG) codes, and Current Procedural Terminology (CPT) codes.
- It is a time series data. Clinical variables are recorded concerning time for each patient. Figure 1 shows the chart of timing clinical variables recorded in the dataset.

The details of datset used in this research is given in table I below.

TABLE I
DATASET SUMMARAY

Data	Statistics
Total Patients	58,576
Patients Selected For Study	6,587
Number of classes	2
Number of patients readmitted in 30 days	2,461
Number of patients readmitted after 30 days	4,126
Number of female patients	2,916
Number of male patients	3,671

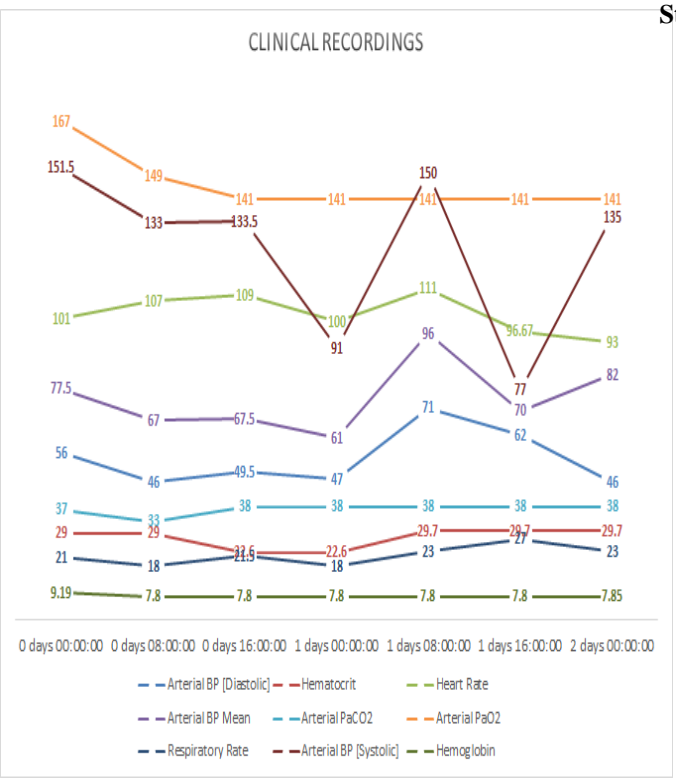

Fig. 1. Some of the Clinical recordings associated with each patient[20]

B. Flow Diagram

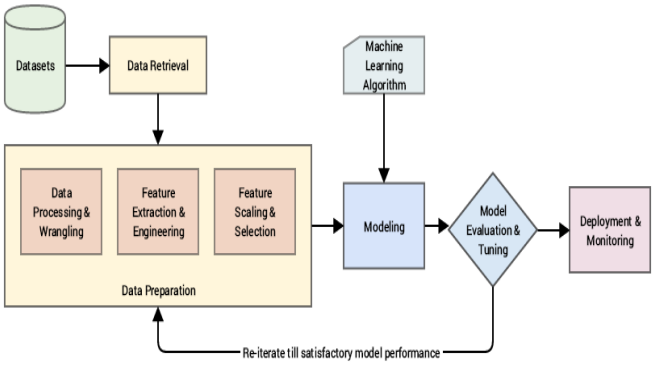

Fig. 2. Flow Diagram

Work flow of methodologies are given below:

Step 1: Collect a vast readmission dataset with several features.
Step 2: Do data-preprocessing to deal with missing values.
Step 3: Convolution layers can be used for feature extraction, i.e., to retrieve the meaningful data out of the dataset.
Step 4: Do Feature scaling to normalize the dataset to reduce biases.
Step 5: Feature selection can also be done using nature-inspired algorithms to fetch important features from the dataset.
Step 6: Feeding the retrieved dataset into the model.
Step 7: Calculate the Accuracy, FScore, AUC, etc. depend on the actual and predicted value.
Step 8: If the results are satisfied, then stop else goto step 2 and redo the procedure with different hyperparameters, algorithms, etc.

C. Data Preprocessing

1) Data Extraction: MIMIC-III dataset consisted of around 58,576 patients who were diagnosed as suffering from different types of disease and hence mortality due to those diseases. These diseases include Pulmonary disease, Circulatory disease, Trauma, a disease of the digestive system and many more. Since the dataset is very large, we only consider data of those patients who were readmitted again which gives the details of 6,587 patients. The data set is divided into two classes:-

- Patients readmitted in 30 days.
- Patients readmitted after 30 days.

2) Normalization: Normalization is used to reduce the biases among the attributes. It presents the attributes on a common scale. It standardizes the span of independent attributes or variables of data, called feature normalization. We use min-max normalization[21] here.

- Min-Max Scaling
  Let a matrix "M":-

$$M = \begin{Vmatrix} m11 & m12 \\ m21 & m22 \end{Vmatrix} \quad (1)$$

So for scaling we compute

$$m = min(m11, m21), \tag{2}$$

$$n = max(m11, m21) \tag{3}$$

$$o = min(m12, m22), \tag{4}$$

$$p = max(m12, m22) \tag{5}$$

And "M" becomes:-

$$M = \begin{Vmatrix} (m11 - m)/(n - m) & (m12 - o)/(p - o) \\ (m21 - m)/(n - m) & (m22 - o)/(p - o) \end{Vmatrix} \tag{6}$$

D. Feature Selection

Features are selected through Nature Inspired Algorithms:

1) Bat Algorithm: Bat Algorithm is an optimization algorithm that mimics the echolocation behavior of microbats[22]. Bats uses echolocation to distinguish between food/prey and they also sense distance and other background barriers. Bats fly at random with a speed u_i at point x_i with a constant frequency μ_{min}, changing wavelength and loudness L_0 to look for the victim. They can immediately change the frequency of their released pulses and regulate the rate of pulse ejection r in the span of [0, 1], based on how close is their prey. Even tough the loudness can alter in different techniques, we suppose that the loudness changes from a high (positive) L_0 to a least fixed value L_{min}. Rules to update position x_i and speed u_i of bats at time step t is given by:

$$\mu_i = \mu_{min} + (\mu_{max} - \mu_{min})\beta \tag{7}$$

$$u_i^t = u_i^{t-1} + (x_i^{t-1} - x_b)\mu_i \tag{8}$$

$$x_i^t = x_i^{t-1} + u_i^t \tag{9}$$

where β $\in$[0,1] is a arbitrary vector extract from a rectangular distribution and after comparing all the solution of n bats we found x_b the global best solution. For the local search part a local arbitrary walk is used to find a new result for each bat once a result is chosen from the current best solution:

$$x_{new} = x_{old} + \epsilon L^t \tag{10}$$

where ϵ $\in$[-1,1] is an arbitrary number. Average loudness of all the n bats at that time is L^t.
Variation of Loudness and Pulse Emission:

$$L_i^{t+1} = \alpha L_i^t \tag{11}$$

$$r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \tag{12}$$

where α & γ are constant.

2) Grey Wolf Optimizer: Grey Wolf Optimizer(GWO) Algorithm imitate the nature of grey wolves[23]. The GWO algorithm imitates the leadership pecking order and coursing procedure of grey wolves in nature. Four sorts of grey wolves are chosen for duplicate the initiative hierarchy such as alpha, beta, delta, and omega. Also, the three primary advances of coursing, hunting for prey, encircling prey, and attacking victim, are implemented.
Alpha wolf is a leader male or female. They make decisions like the sleeping place, hunting, etc. Other wolves acknowledge alpha wolf by their tail down. Beta wolf help alpha wolf in making decisions. They are an advisor for alpha and discipliner for the pack. They also ensure all subordinate obey the order of alpha and give feedback to alpha. Delta wolves also called subordinate. They dominate omega wolves.
Categories of Delta wolves:-

- Scouts - Watch boundaries.
- Sentinels - Protect pack.
- Elders - Aplha or beta wolf sometimes.
- Hunters - Help aplha and beta wolf in hunting.
- Care Taker - Care ill weak and wounded wolves.

Omega wolves are like the scapegoat in the pack. They are having weak fitness and are allowed last to eat.
Search Process in GWO Algorithm:-

- Searching
- Encircling
- Attacking the prey

a) Searching:

- Search according to aplha, beta and delta wolves. They separate to look for prey and come together to ambush on prey.
- Modeled by using w random variable greater than 1 or less than -1.
- When $|w| > 1$ wolves are required to separate from prey to find better solution.

b) Encircling Prey: It is the process in which wolves encircle the prey for attack. It is modeled as:

$$\overrightarrow{\mathrm{d}} = |\overrightarrow{\mathrm{c}}\,\overrightarrow{\mathrm{x}}_p(t) - \overrightarrow{\mathrm{x}}(t)| \tag{13}$$

$$\overrightarrow{\mathrm{x}}(t+1) = \overrightarrow{\mathrm{x}}_p(t) - \overrightarrow{\mathrm{w}}\,\overrightarrow{\mathrm{d}} \tag{14}$$

in which t is the ongoing loop, $\overrightarrow{\mathrm{w}}$, $\overrightarrow{\mathrm{c}}$ are coefficient vectors and $\overrightarrow{\mathrm{x}}_p$ is location of prey and $\overrightarrow{\mathrm{x}}$ is the location of grey wolf.

Updation of Coefficients

$$\overrightarrow{\mathrm{w}} = 2\overrightarrow{\mathrm{a}}\,\overrightarrow{\mathrm{r1}} - \overrightarrow{\mathrm{a}} \tag{15}$$

$$\overrightarrow{\mathrm{c}} = 2\overrightarrow{\mathrm{r2}} \tag{16}$$

Where $\overrightarrow{\mathrm{a}}$ linearly decrease from two to zero over the course of iteration. $\overrightarrow{\mathrm{r1}}$ and $\overrightarrow{\mathrm{r2}}$ are arbitrary vectors $\in$ [0,1].

c) Attacking: The attack is mostly led by alpha. The beta and delta may also join in hunting. We first save three best solutions i.e. alpha, beta and delta which are the best answers and update the locations of other wolves based on these three. It is modeled as:

$$\overrightarrow{\mathrm{d}}_{\alpha} = |\overrightarrow{\mathrm{c}}_1 \overrightarrow{\mathrm{x}}_{\alpha} - \overrightarrow{\mathrm{x}}| \tag{17}$$

$$\overrightarrow{\mathrm{d}}_{\beta} = |\overrightarrow{\mathrm{c}}_2 \overrightarrow{\mathrm{x}}_{\beta} - \overrightarrow{\mathrm{x}}| \tag{18}$$

$$\overrightarrow{\mathrm{d}}_{\delta} = |\overrightarrow{\mathrm{c}}_3 \overrightarrow{\mathrm{x}}_{\delta} - \overrightarrow{\mathrm{x}}| \tag{19}$$

$$\overrightarrow{\mathrm{x}}_1 = \overrightarrow{\mathrm{x}}_{\alpha} - \overrightarrow{\mathrm{w}}_1 \overrightarrow{\mathrm{d}}_{\alpha} \tag{20}$$

$$\overrightarrow{\mathrm{x}}_2 = \overrightarrow{\mathrm{x}}_{\beta} - \overrightarrow{\mathrm{w}}_2 \overrightarrow{\mathrm{d}}_{\beta} \tag{21}$$

$$\overrightarrow{\mathrm{x}}_3 = \overrightarrow{\mathrm{x}}_{\delta} - \overrightarrow{\mathrm{w}}_3 \overrightarrow{\mathrm{d}}_{\delta} \tag{22}$$

$$\overrightarrow{\mathrm{x}}(t+1) = (\overrightarrow{\mathrm{x}}_1 + \overrightarrow{\mathrm{x}}_2 + \overrightarrow{\mathrm{x}}_3)/3 \tag{23}$$

Grey wolf attack it's target when it is immobile. In GWO vector w is a irrational value within an range [-2a,2a] a decrease from two to zero throughout the iteration. When $|w| < 1$ wolves ambush on victim.

E. Feature Extraction

For feature extraction Convolution Neural Network(CNN) is used.

1) Convolutional Neural Network: A convolution neural network(CNN)[24] is an efficient machine learning approach from deep learning and is close to standard multi layered networks. It consists of three layers:

- Convolution Layer
- Pooling Layer
- Fully Connected Layer

Convolutional layers: This layer is the most important layer among all in CNN. It contains rectangular grids or cubic blocks called filters. The filter is moved at every block of input neuron to produce output. These filters are trainable. Output is determined by equation:

$$WI^x + b \tag{24}$$

There are three hyperparametrs in convolutionlayer that decides the output:

- Depth
- Stride
- Zero Padding

The number of filters used in convolution layer with certain stride is the depth of the output neuron. For Example say, the RGB image, with a depth of three. Given an image of size (5x5x3) and given ten filters of size (3x3x3). Here, when the value of stride is given 1, the filters hovers a single pixel over an image then the output neuron is of size (3x3x10). Whilst when the value of stride is given 2, the filters will skip two pixels, and output neuron is of size (2x2x10).

Zero padding is the process of filling zeros around the edge of the input neuron. Zero padding is commonly used to adjust the size of input neuron during the filter process when we need to main the output neuron based on input neuron size.

$$Output = (W - F + 2P)/S + 1 \tag{25}$$

in which the input neuronś size is W, filterś (kernel) size is F,paddingś size is P, and stride is S. Linear algebra are also used in the convolutional neural network. Now consider that given a matrix of dimension AxB (A is the count of rows and B is the count of columns). Now two dimensional convolution operation with two matrices (A_M, B_M) is a dimension of matrix M, and (A_N, B_N) is a dimension of matrix N. Convolution of these matrix is given by below equation:

$$P(i,j) = \sum_{a=0}^{A_M-1} \sum_{b=0}^{B_M-1} M(a,b) * N(i-a, j-b) \tag{26}$$

Max-pooling layer: Pooling layer executes the next operation after each convolution layer. These layers are utilized to minimize the size of the neurons. These are small rectangular grids that acquires small portion of convolutional layer and filters it to give a result from that block. The most commonly used layer is max pooling that fetch that maximum pixel from the block. A expression for a max-pooling layer is expressed in a equation:

$$h_l^j(a,b) = max \in \mathcal{N}(a), b \in \mathcal{N}(b) h_j^{l-1}(\bar{x}, \bar{b}) \tag{27}$$

Fully Connected layers: The final layer of a convolutional neural network(CNN) is a fully connected layer that is formed from the attachment of all preceding neurons. It reduces the spatial information as it is fully connected like in artificial neural network. It contains neurons beginning at input neurons to the output neurons.

Details of each layer and their details are presented in Table II.

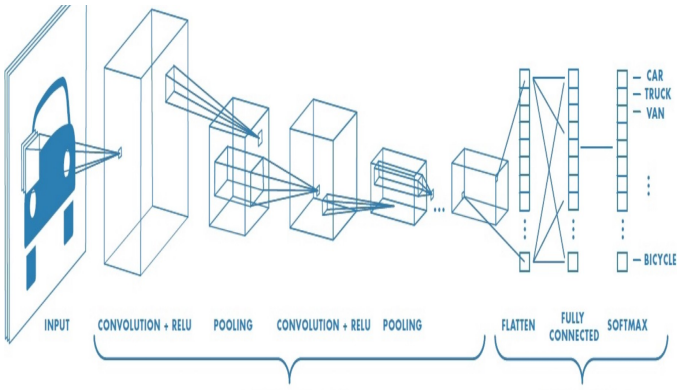

Fig. 3. Convolution Neural Network Architecture

TABLE II
CONVOLUTION NEURAL NETWORK SUMMARY

Layers	Kernel Size	Filters Used	Input Shape	Output Shape
Conv1D	5	32	(98,30)	(98,32)
Conv1D	5	32	(98,32)	(98,32)
Conv1D	5	64	(98,32)	(98,64)
MaxPooling1D	2	1	(98,64)	(98,32)

TABLE III
FEATURES SELECTED BY DIFFERENT ALGORITHMS USED

Algorithms	Number of Features Selected
Bat Algorithm	21
Grey Wolf Optimizer	17
Convolution Neural Network	32

F. Algorithms

Recurrent neural networks (RNN) are utilized to represent the time dependency in the time series data[25]. As most of the healthcare data is a series of temporal recordings, hence RNNs have been widely adopted in the healthcare domain. We are usually provided with a series of observations $x_1 ... x_T$ and we train a classifier to generate hypotheses ŷ. The recurrent connections are added in feed-forward neural networks to make it RNN. The output of a neuron in a typical NN is as follows:

$$y_i^t = \sigma(W_i x^t + b_i) \tag{28}$$

Where the weight matrix is represented as W_i, the bias represented as b_i and σ represents the sigmoid function. While in the case of RNN, a neuron is fed with the result of the neuron at time $t-1$. The following equation shows the new activation function:

$$y_i^t = (W_i x^t + V_i x^{t-1} + b_i) \tag{29}$$

As RNN uses the previous outputs as recurrent connection, their current output depends upon the previous states. This property of RNN makes it very useful in sequence labeling tasks. The backpropagation through time can be used to train RNNs. It was demonstrated by that learning long-term dependencies is difficult using gradient descent. This is mainly because the backpropagating error can vanish which makes the network inefficient in learning long-range dependencies, or frequently explode which makes convergence impossible. LSTM models were designed to tackle the concern of vanishing gradients and were developed to model long-range dependencies efficiently. LSTMs can fix this by maintaining an internal state that shows the memory cell of the LSTM neuron. This internal state can only be written and read via gates which decides what information will through gates. The following diagram shows the recurrent neural network.

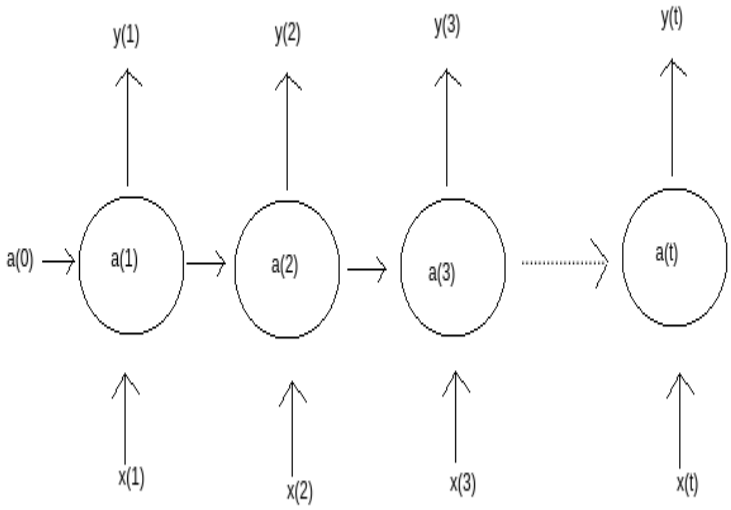

Fig. 4. An recurrent neural network

1) Long Short Term Memory (LSTM) Networks: Long Short Term Memory networks are commonly called LSTMs. They are a particular kind of RNN, that are good in understanding long-term dependencies. They were originated by [26] and were polished and familiarized by many people. They work very effectively on a large diversity of problems and are now used by many researchers. To solve long-term dependency problem, LSTMs are precisely formulated. Their default behaviour is to remembering something for a long duration of time. All recurrent neural networks contains a series of replicating chunks of the neural network. In traditional RNNs, this replicating chunk have a straightforward architecture, such as a single tank layer. LSTMs also have this series-like architecture, but the recurrent chunk has a disimilar architecture. Rather than having a single layer, there are four layers which are connecting extraordinarily. The intermediate information is stored in a single hidden layer l and its state changes over time (l_{t-1}, l_t, l_{t+1}). On the final hidden state vector l_T, we used a fully connected layer followed by sigmoid function. For loss function, we used log loss. The following equations can be used for calculation of the current hidden layer l_t.

$$f_t = \sigma(W_f X_t + R_f l_{t-1} + b_f) \tag{30}$$

$$u_t = \sigma(W_u X_t + R_u l_{t-1} + b_u) \tag{31}$$

$$o_t = \sigma(W_o X_t + R_o l_{t-1} + b_o) \tag{32}$$

$$\tilde{C}_t = \Phi(W_C X_t + R_C l_{t-1} + b_c) \tag{33}$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \tag{34}$$

$$l_t = o_t \phi(C_t) \tag{35}$$

u_t, o_t and f_t are input gate, output gate and forget gate respectively. Forget gate decides which information to discard, input gates are used to update cell and output gates are used to decide the output of the cell state. Cell states are completely overridden in classical RNN, but LSTM has the potential to eliminate or append data to the cell state. The input weight of each gate, recurrent weight, and the bias are expressed as W* , R* , b* respectively where * can be f, u, o and c. Here σ, Φ stands for an element wise operation of the sigmoid (logistic) and tanh function respectively. For matrix multiplication. The candidate values are computed in equation(3.33), and equation(3.34) old state is multiplied by f_t, and this helps in neglecting the things we decided to neglect. Then we add u_t*δC_t in it. This is the new candidate values, expanded by the amount we wanted to reform each state value. The final output of an LSTM unit is given by equation(3.35). Here * represents the Hadamard (element-wise) multiplication process. Figure 3 shows the LSTM network.

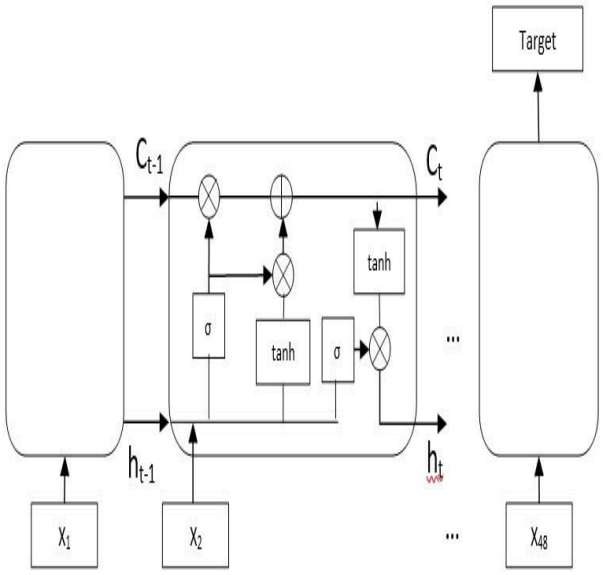

Fig. 5. A LSTM Network

2) Bidirectional Recurrent Neural Networks (BRNN): Bidirectional Recurrent Neural Networks also known as BRNN is just like RNN but it trains simultaneously on both directions of the time dependent data[27]. This algorithm gives the better result in both regression and classification problem. BRNN computes both forward ($\overrightarrow{\mathrm{l}}$) and backward ($\overleftarrow{\mathrm{l}}$) hidden sequence.

$$\overrightarrow{\mathrm{l}}_t = \mathcal{O}(W_{x\overrightarrow{\mathrm{l}}} x_t + W_{\overrightarrow{\mathrm{l}}\overrightarrow{\mathrm{l}}} \overrightarrow{\mathrm{l}}_{t-1} + b_{\overrightarrow{\mathrm{l}}}) \tag{36}$$

$$\overleftarrow{\mathrm{l}}_t = \mathcal{O}(W_{x\overleftarrow{\mathrm{l}}} x_t + W_{\overleftarrow{\mathrm{l}}\overleftarrow{\mathrm{l}}} \overleftarrow{\mathrm{l}}_{t+1} + b_{\overleftarrow{\mathrm{l}}}) \tag{37}$$

$$y_t = W_{\overrightarrow{\mathrm{l}}y} \overrightarrow{\mathrm{l}}_t + W_{\overleftarrow{\mathrm{l}}y} \overleftarrow{\mathrm{l}}_t + b_o \tag{38}$$

The long range context can be accessed in both directions by combining Bid-diretional RNNs with LSTM which gives bidirectional LSTM[28].

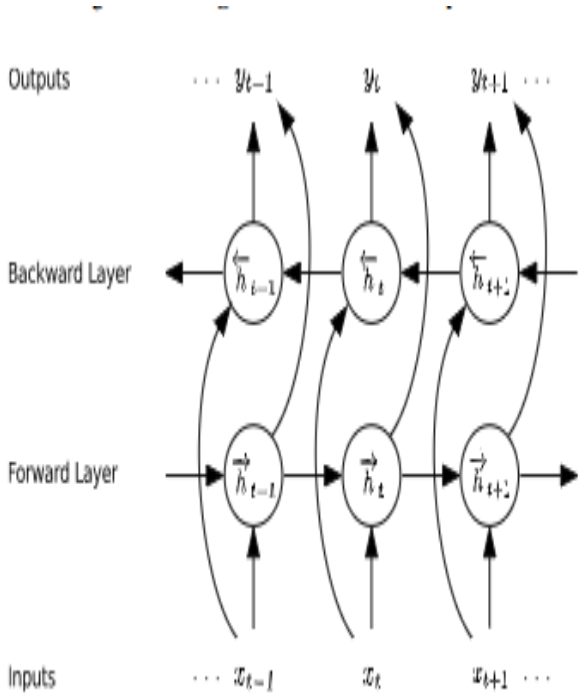

Fig. 6. Bidirectional Recurrent Neural Networks

3) Auto Encoders: An auto-encoder[29] is a neural network model that seeks to learn a flatten representation of the data. They are an unsupervised learning algorithm, although commonly, they are trained using supervised learning algorithms, mentioned as self-supervised. They are commonly trained as part of a deeper model that attempts to recreate the input. The design of the auto-encoder model purposefully makes this challenging by limiting the architecture to a bottleneck at the center of the model, from which the reconstruction of the input data is performed. In this case, once the algorithm is fit, the regeneration aspect of the model can be dropped, and the model up to the point of the bottleneck can be adopted. The result of the model at the congestion is a fixed length vector that produces a compressed representation of the input data. Input data from the region can then be supplied to the model, and the result of the model at the congestion can be used as a attribute vector in a supervised learning model, for visualization, or more generally for reducing dimensions.

IV. Results

A. Implementation Details

The entire dataset was split using stratified sampling and 10% of the dataset was used as a validation set and random search was used for hyperparameter optimization. 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.
Another model train on this dataset was convolution LSTM (CLSTM). It is a hybrid model in which convolution layers are used for feature extraction. The CLSTM model was trained with 30 epochs and the bunch 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.
In the training phase, the model was saved and also compute these performance metrics (AUC, F1, precision, and recall).
Several built-in modules of python will be utilized, namely:

- Numpy
- Pandas
- Scikit-Learn
- Keras
- Matplotlib
- NiaPy

The computations were performed on Ubuntu 16.04 LTS with following hardware specifications:

- **Processor** Intel i7 6th Gen. broadwalle processor.
- **RAM** 24 GB DDR4
- **Secondary Memory** 250 GB SSD
- **Graphics Card** Nvidia 1080Ti 11GB

B. Results Obtained

This segment shows the readmission prediction results acquired via various predictive algorithms. The prediction accuracy will be differentiate on the basis of two variables final accuracy and Area under Receiver Operating Characteristic Curve (AUC-ROC)[30]. Both final accuracy and AUC-ROC curve relies on the classifierś capability to grade examples for positive class, however in the case of final accuracy, it further relies on the capability to compute the edge in the ordering to distinguish the positive class. Applying various models on the data, with default attributes, we got results as referenced in Table IV. Also we compare deep learning results with baseline models in Table V to show how deep learning helps in improving the accuracy. AUC-ROC curves and Model Loss curves are shown from figure 7 to figure 13 and from figure 15 to figure 17 respectively.

1) Curves showing Receiver Operating Curve (ROC) and Loss Curve of different models:

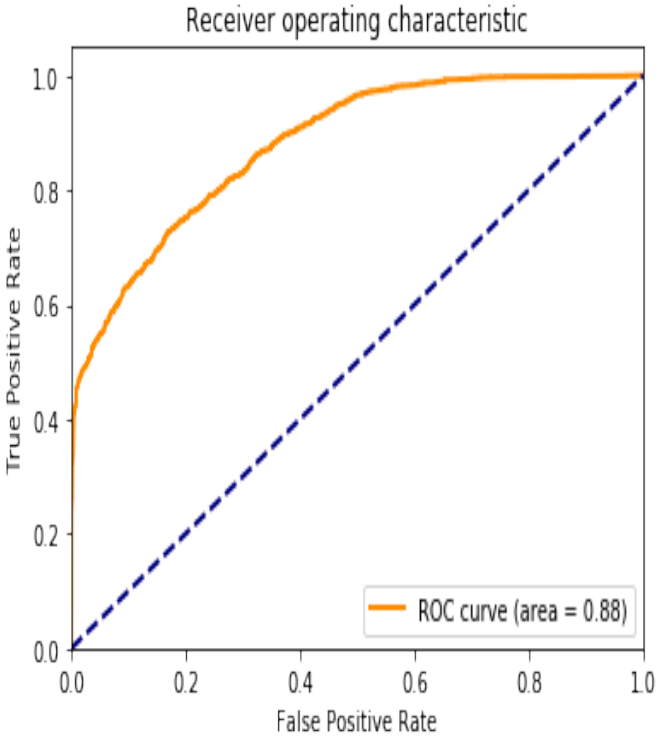

Fig. 7. ROC Curve: LSTM.

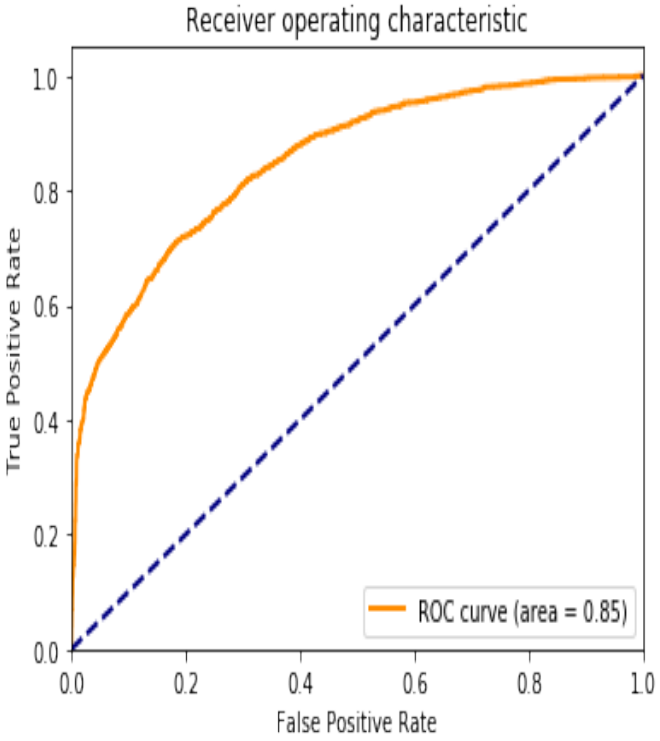

Fig. 8. ROC Curve Bi-Directional LSTM.

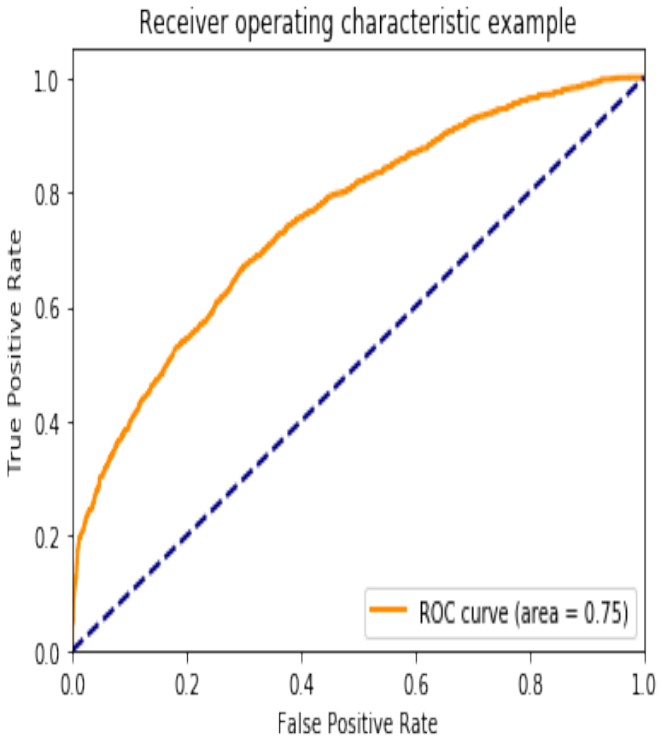

Fig. 9. ROC Curve: Convolution LSTM.

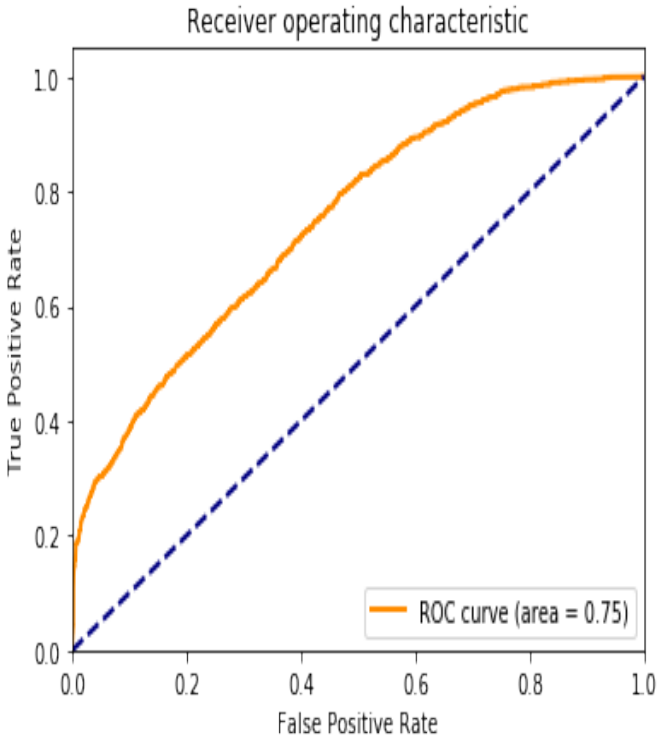

Fig. 10. ROC Curve: LSTM with Bat Algorithm.

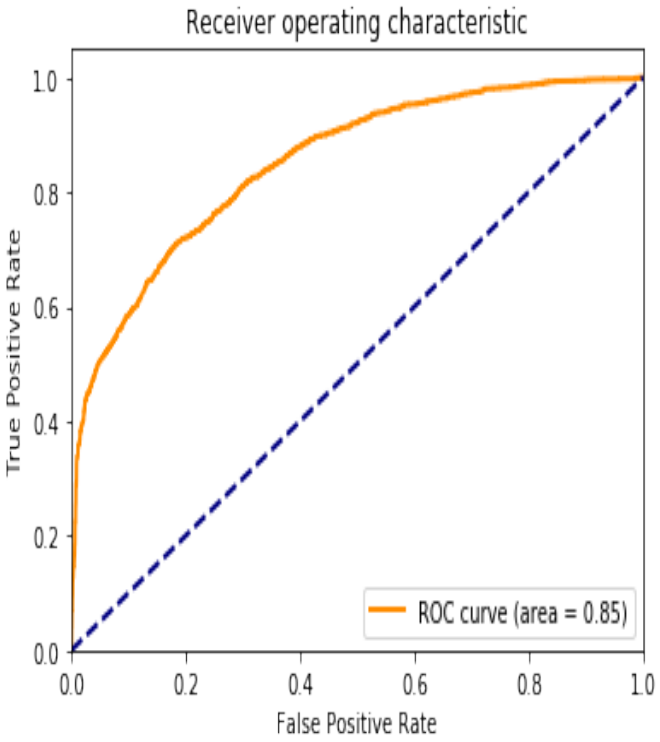

Fig. 11. ROC Curve: LSTM with Grey Wolf Optimizer Algorithm.

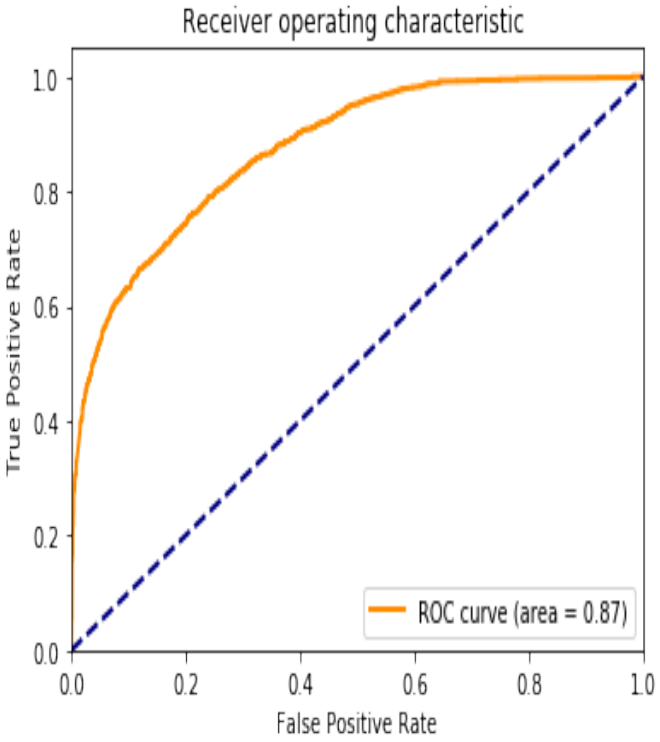

Fig. 12. ROC Curve Convolution LSTM Auto Encoders.

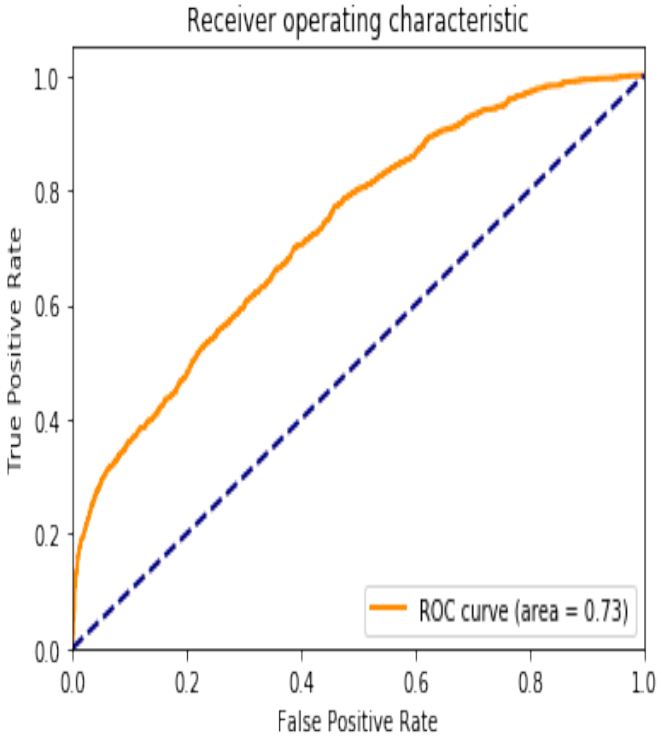

Fig. 13. ROC Curve: LSTM Auto-Encoders.

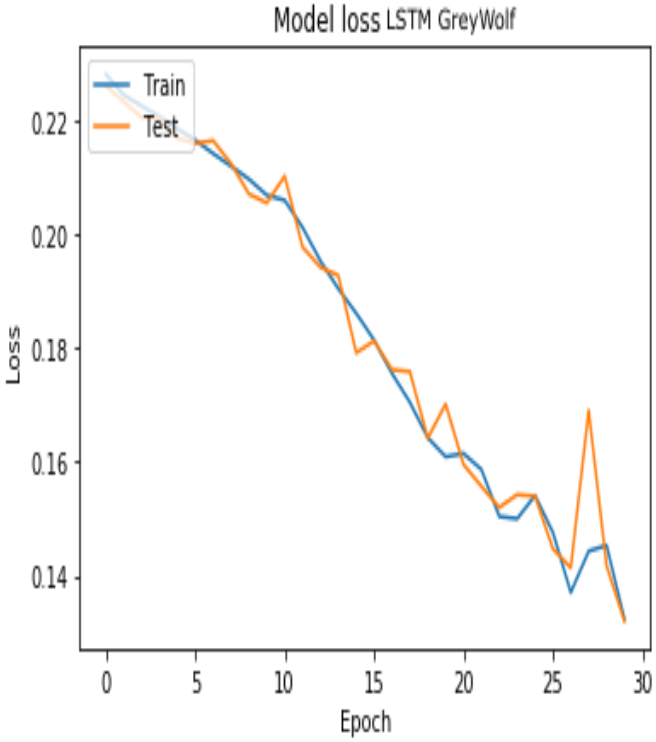

Fig. 15. Loss Curve: LSTM Grey Wolf.

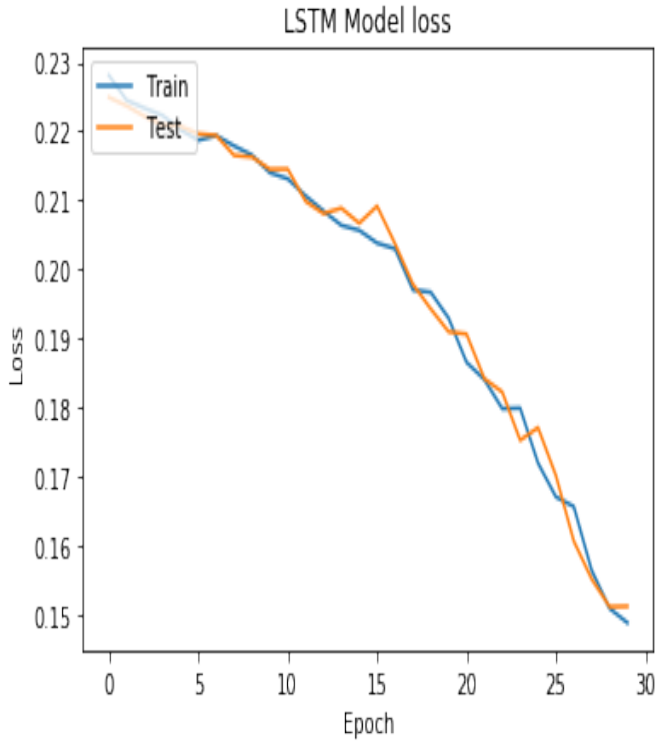

Fig. 14. Loss Curve: LSTM.

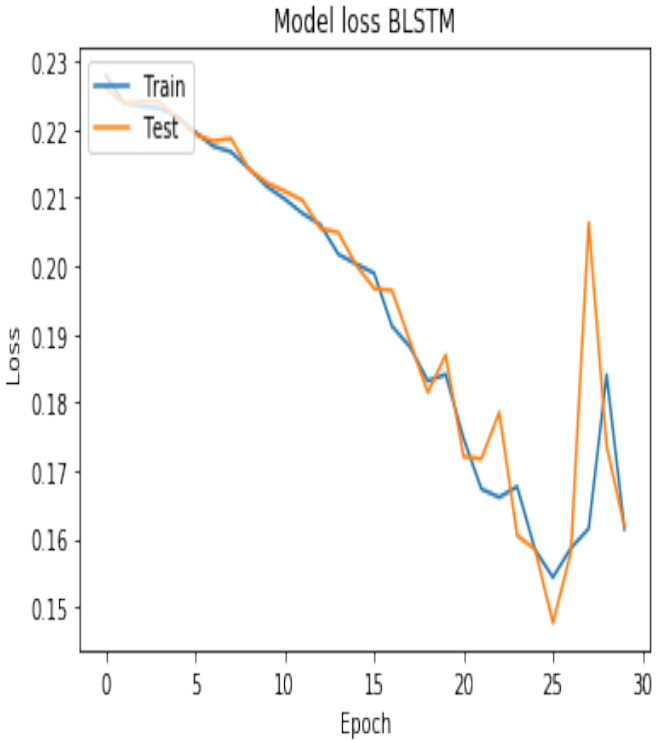

Fig. 16. Loss Curve: Bidirectional LSTM.

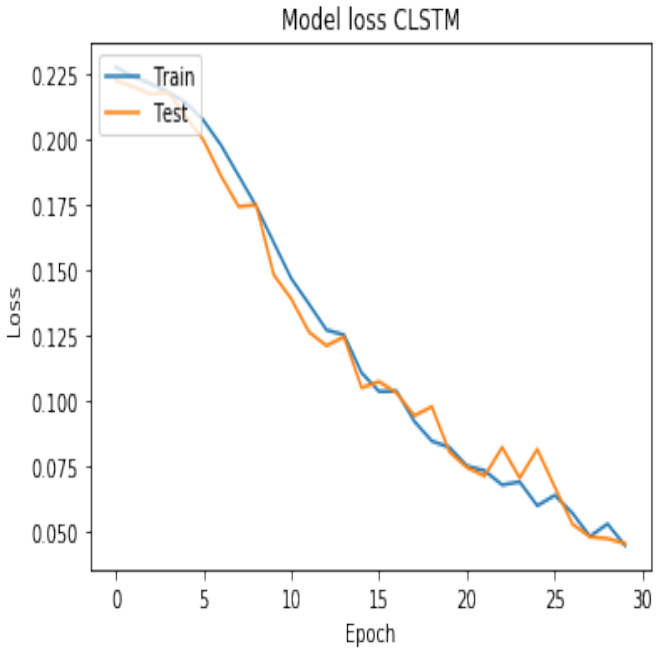

Fig. 17. Loss Curve: Convolution LSTM.

V. Conclusion

Halth Care Readmissions is extremely difficult task for both sufferer and health center. In our thesis, we design a robust algorithm to learn the amount of patients admitted again to a health care. We measured the performance of different deep learning algorithms with and without Nature-inspired algorithms to predict readmission probability and concluded that Long Short Term Memory with grey wolf optimizer performed better than the remaining ML algorithms in the prediction value. We also establish that the result of the combination of LSTM and Convolution layer was remarkable on this dataset. This architecture can also be used in current's health system to aim at high possibility patients, decrease the degree of readmission, and provide excellent health care.

References

[1] Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J. and Clore, J. N.: 2014, Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records, BioMed research international 2014.

[2] Bengio, Y. et al.: 2009, Learning deep architectures for ai, Foundations and trends R in Machine Learning 2(1), 1127.

[3] Lipton, Z. C.: 2015, A critical review of recurrent neural networks for sequence learning, CoRR abs/1506.00019.

[4] Hochreiter, S. and Schmidhuber, J.: 1997, Long short-term memory, Neural computation 9(8), 17351780.

[5] Ghassemi, M. M., Richter, S. E., Eche, I. M., Chen, T. W., Danziger, J. and Celi, L. A.: 2014, A data-driven approach to optimized medication dosing: a focus on heparin, Intensive care medicine 40(9), 13321339.

[6] Pirracchio, R., Petersen, M. L., Carone, M., Rigon, M. R., Chevret, S. and van der Laan, M. J.: 2015, Mortality prediction in intensive care units with the super icu learner algorithm (sicula): a population-based study, The Lancet Respiratory Medicine 3(1), 4252.

[7] Che, Z., Purushotham, S., Khemani, R. G. and Liu, Y.: 2015, Distilling knowledge from deep networks with applications to healthcare domain, CoRR abs/1512.03542.

[8] Liang, M. and Hu, X.: 2015, Recurrent convolutional neural network for object recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 33673375.

[9] Ioffe, S. and Szegedy, C.: 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift, Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML15, JMLR.org, pp. 448456. URL: http://dl.acm.org/citation.cfm?id=3045118.3045167

[10] Che, Z., Purushotham, S., Khemani, R. and Liu, Y.: 2016, Interpretable deep models for icu outcome prediction, AMIA Annual Symposium Proceedings, Vol. 2016, American Medical Informatics Association, p. 371.

[11] Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F. and Sun, J.: 2016, Doctor ai: Predicting clinical events via recurrent neural networks, Machine Learning for Healthcare Conference, pp. 301318.

[12] Choi, E., Schuetz, A., Stewart, W. F. and Sun, J.: 2016, Using recurrent neural network models for early detection of heart failure onset, Journal of the American Medical Informatics Association 24(2), 361370.

[13] Hanson, J., Yang, Y., Paliwal, K. and Zhou, Y.: 2016, Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks, Bioinformatics 33(5), 685692.

[14] Che, Z., Sauver, J. S., Liu, H. and Liu, Y.: 2017, Deep learning solutions for classifying patients on opioid use, AMIA Annual Symposium Proceedings, Vol. 2017, American Medical Informatics Association, p. 525.

[15] Dey, R. and Salemt, F. M.: 2017, Gate-variants of gated recurrent unit (gru) neural networks, Circuits and Systems (MWSCAS), 2017 IEEE 60th International Midwest Symposium on, IEEE, pp. 15971600.

[16] Greff, K., Srivastava, R. K., Koutnk, J., Steunebrink, B. R. and Schmidhuber, J.: 2017, Lstm: A search space odyssey, IEEE transactions on neural networks and learning systems 28(10), 22222232.

[17] Rubin, J., Abreu, R., Ganguli, A., Nelaturi, S., Matei, I. and Sricharan, K.: 2017, Recognizing abnormal heart sounds using deep learning, KHD@IJCAI.

[18] Franco, J., Formiga, F., Trullas, J.-C., Bautista, P. S., Conde, A., Manzano, L., Quiros, R., Franco, A. G., Ezquerro, A. M., Montero-Perez-Barquero, M. et al.: 2017, Impact of prealbumin on mortality and hospital readmission in patients with acute heart failure, European journal of internal medicine 43, 3641.

[19] Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H. and Adeli, H.: 2018, Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals, Computers in biology and medicine 100, 270278

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

[21] Jain, Y. Kumar, and Santosh Kumar Bhandare. "Min max normalization based data perturbation method for privacy protection." International Journal of Computer & Communication Technology 2.8 (2011): 45-50.

[22] Yang, X.-S. and Hossein Gandomi, A.: 2012, Bat algorithm: a novel approach for global engineering optimization, Engineering Computations 29(5), 464483.

[23] Lipton, Z. C.: 2015, A critical review of recurrent neural networks for sequence learning, CoRR abs/1506.00019.

[24] Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." arXiv preprint arXiv:1404.2188 (2014).

[25] Che, Z., Purushotham, S., Cho, K., Sontag, D. and Liu, Y.: 2018, Recurrent neural networks for multivariate time series with missing values, Scientific reports 8(1), 6085.

[26] Hochreiter, S. and Schmidhuber, J.: 1997, Long short-term memory, Neural computation 9(8), 17351780.

[27] Schuster, M. and Paliwal, K. K.: 1997, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45(11), 26732681.

[28] Graves, A. and Schmidhuber, J.: 2005, Framewise phoneme classification with bidirectional lstm and other neural network architectures, Neural Networks 18(5- 6), 602610.

[29] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. 2019.

[30] Rubin, J., Abreu, R., Ganguli, A., Nelaturi, S., Matei, I. and Sricharan, K.: 2017, Recognizing abnormal heart sounds using deep learning, KHD@IJCAI.