Hospital Readmission Prediction Of ICU Patients Using Deep Learning Algorithms

by

Shivam Sinha

2014IPG-082

A thesis submitted in partial fulfilment of the requirements for the award of the degree of

Master of Technology

in

Information Technology

2018-19

विश्वजीवनामृतं ज्ञानम्

ATAL BIHARI VAJPAYEE-

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015, MADHYA PRADESH, INDIA

Thesis Certificate

I hereby certify that the work, which is being presented in the report/thesis, entitled Hospital Readmission Prediction Of ICU Patients Using Deep Learning Algorithms, in fulfillment of the requirement for the award of the degree of **Master of Technology** in **Information Technology** and submitted to the institution is an authentic record of my own work carried out during the period *May-2018* to *July-2019* under the supervision of Dr. Ritu Tiwari. I also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Dr. Ritu Tiwari

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Candidate's Declaration

I hereby certify that I have properly checked and verified all the items as prescribed in the check-list and ensure that my thesis is in the prop er format as specified in the guideline for thesis preparation.
I declare that the work containing in this rep ort is my own work. I understand that plagiarism is defined as any one or combination of the following:

(1) To steal and pass off (the ideas or words of another) as one's own

(2) To use (another's production) without crediting the source

(3) To commit literary theft

(4) To present as new and original idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else's work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else's work constitute plagiarism.
I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Signature:

Name:

Roll. No:

Date:

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. Healthcare has shown promises, as there is a huge share of the budget in intensive care medicine. This has expanded the interests of healthcare researchers, and they are focusing on providing the best medication to critically-ill patients. DNN models are advanced machine learning models, which have been widely applied for clinical applications in recent times. Due to the lack of ability to capture time dependency in the data which is a major shortcoming, basic machine learning algorithms like Logistic Regression, SVM, Decision Tree, etc. are not very useful. There are several types of deep learning algorithm like Deep belief networks, Deep convolutional neural network, Recurrent neural network, etc. The LSTM variant of the Recurrent neural network have shown promises to work better on data involving time series prediction, and it can transfer the results of previous cell state onto the next cell state so that dependency on time is maintained throughout. In this thesis, the proposed methodology is to use LSTM, and its different variants like Gated Recurrent Unit (GRU), LSTM attention on healthcare applications like hospital readmission prediction using ICU data. The different architectures of LSTM will produce different results based on the data on which architecture is being trained. Then the comparison of performances can be done between LSTM, and it's variants on various healthcare applications and provide the best model suitable for each of the different healthcare data being tested. The comparison of time independent machine learning models like Logistic Regression, SVM, Naive Bayes can be done with the results of Deep learning models.

Keywords: Deep learning, LSTM, Healthcare, Recurrent Neural Network, GRU.

Acknowledgments

I am highly indebted to **Dr. Ritu Tiwari**, and obliged for giving me the autonomy of functioning and experimenting with ideas. I would like to take this opportunity to express my profound gratitude to him not only for his academic guidance but also for his personal interest in my report and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within me. The nurturing and blossoming of the present work is mainly due to his valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. My mentor always answered myriad of my doubts with smiling graciousness and prodigious patience, never letting me feel that I am novices by always lending an ear to my views, appreciating and improving them and by giving me a free hand in my report. It's only because of his overwhelming interest and helpful attitude, the present work has attained the stage it has.

Finally, I am grateful to our Institution whose constant encouragement served to renew my spirit, refocus my attention and energy and helped me in carrying out this work.

Contents

Tables

Table

Figures

Figure

Chapter 1

Introduction

This chapter offers an overview of the context as a part of Hospital Readmission project and how machine learning used in health care applications in section 1.1 and 1.2, respectively. In section 1.3, 1.4, and 1.5, we discuss the algorithms used in this project. In section 1.6, the problems and motivations are presented. Next, in section 1.7, we discuss the objectives of this project.

1.1 Hospital Readmission

Medical clinic Readmissions are distinguished as a sign of the low quality of consideration, for example, inadequate release arranging and care coordination. Emergency clinic readmission[32] happens when a patient release from a medical clinic is readmitted inside a predetermined interim. Readmission can happen for any decided or undecided reason. These elements drove Hospital associations to consider controlling 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 quality benchmark for wellbeing frameworks. For Medicare patients, hospitalizations are exceptionally distressing, and it is considerably more when they result in ensuing readmissions. 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.

1.2 Machine Learning For Health Care

Machine learning leads to provide intelligence by providing new knowledge. Since the onset of machine learning, it has been widely used 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. Machine learning can assist healthcare providers in a variety of tasks like hospital readmission of patients suffering from chronic illness. 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.

1.3 Deep Learning

Deep learning is one of the extended section of Machine learning, which works on a specific machine learning 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].

1.4 Recurrent Neural Network

An artificial neural network has a specific king 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 is alterable continuous-valued weight in each connection. There are input, output, and hidden nodes. 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[25].

1.5 Long Short Term Memory

The Long short-term memory (LSTM) network, a deep learning RNN, is used by numerous researchers, published by Schmidhuber & Hochreiter in 1997[19]. This deep learning system fix the the vanishing gradient issue of standard RNN. The recurrent gates called forget gates are introduced in LSTM. The vanishing or exploding of back propagated error is prevented in LSTM-RNNs. 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 flow of information 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.

1.6 Motivation

- For Medicaid patients, hospitalizations can be distressing, even more so when they occur in frequent readmissions. A number of researches show that health cares can involve in several projects to lower their degree of readmissions, such as simplifying patient release procedure, synchronizing 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.

1.7 Objectives

The chief objective of this research is:

- To design a Deep Neural Network(DNN) models for Healthcare application, i.e., Hospital Readmission.
- To predict hospital readmission of patients using trained deep neural network models.
- To compare results between Deep Learning Models and with existing algorithms.

1.8 Research work flow

This section focuses on how the research is being carried out, how the different algorithms, models and data prepossessing are implemented.

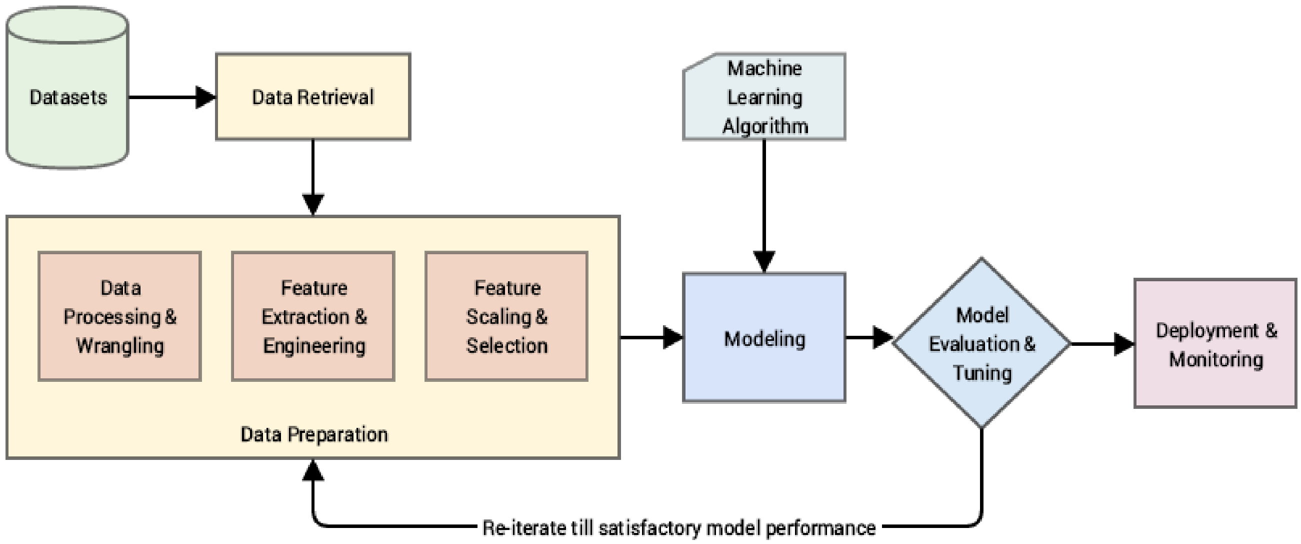

Figure 1.1: Flowchart of proposed methodology

According to the research objectives, the report will describe the work flow as 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. based on the actual and predicted output.

Step 8: If the results are satisfied, then stop else goto step 2 and redo the procedure with different hyperparameters, algorithms, etc.

Chapter 2

Literature review

2.1 Background

A procedure of the current work was performed with a specific end goal to study the most significant healthcare applications literature for the survey and also to study an expansive piece of the most profound learning approaches for healthcare applications.

2.2 Related Research and Analysis

Author (Year)	**Meta-heuristic**	**Paper Name**	**Comments**
Ghassemi et. al (2014) [14]	Multivariate Logistic Regression	A data-driven method to optimized medication dosing: a focus on heparin	An approach is developed that help clinicians determine the optimal initial dose of a drug to safely and quickly reach a therapeutic aPTT window.

Author (Year)	**Meta-heuristic**	**Paper Name**	**Comments**
Pirrachio et al. (2015) [28]	Super ICU Learner Algorithm	Mortality prediction in intensive care units with the Super ICU Learner Algorithm (SICULA): a population-based study.	A super learner algorithm is used for predicting hospital mortality in patients.
Che et al. (2015) [7]	Stacked Denoising Autoencoder & Long Short-Term Memory	Distilling knowledge from deep networks with applications to healthcare domain	A novel called Interpretable Mimic Learning is used, to study interpretable phenotype characteristics for making robust prediction while impersonating the performance of deep learning models meta-heuristics.
Lian and Hu (2015) [24]	Convolution & Recurrent Neural Network	Recurrent convolutional neural network for object recognition	Combination of convolution and recurrent neural network model is used for object detection.
Ioffe and Szegedy (2015) [20]	Batch Normalization	Batch normalization: Accelerating deep network training by reducing internal covariate shift	Reduction in overfitting using batch normalization.

Author (Year)	Meta-heuristic	Paper Name	Comments
Che et. al (2016) [6]	Knowledge Distillation Approach	Interpretable deep models for icu outcome prediction	Interpretable mimic learning is introduced that uses gradient boosting trees to determine interpretable models and at the same time achieves robust prediction performance as deep learning models.
Choi, Bahadori, Schuetz, Stewart and Sun (2016) [9]	Recurrent Neural Networks	Doctor ai: Predicting clinical events via recurrent neural networks	The proportional split of data is found useful for deep learning applications when dealing with the large volume of data.
Choi, Bahadori, Schuetz, Stewart and Sun (2016) [10]	Gated Recurrent Units	Using recurrent neural network models for early detection of heart failure onset	Leveraging longitudinal EHR records for early detection of heart failure using Recurrent Neural Network.
Hanson et al. (2016) [18]	Bidirectional LSTM	Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks	Long short term memory RNN is used to predict disordered proteins, such as SPOT-disorder.

Author (Year)	Meta-heuristic	Paper Name	Comments
Jhonson et al. (2016) [23]	Dataset Extraction	MIMIC-III, a freely accessible critical care database	Medical Information Mart for Intensive Care (MIMIC-III) consists of data about patients admitted to various critical care units in a large hospital.
Che et al. (2017) [8]	Recurrent Neural Networks	Deep Learning Solutions for Classifying Patients on Opioid Use	State-of-the-art deep and recurrent neural network models were applied to achieve robust results on classifying opioid users.
Dey and Salemt (2017) [11]	Gated Recurrent Unit	Gate-variants of Gated Recurrent Unit (GRU) neural networks	Unlock the vanishing gradient problem of a standard RNN.
Greff et al. [17]	LSTM	LSTM: A search space odyssey	Overcomes the vanishing gradient problem of conventional RNN
Rubin et al. (2017) [30]	Convolution Recurrent neural Network	Recognizing abnormal heart sounds using deep learning	Merges the use of time-frequency heat map representations with a deep convolutional neural network (CNN) to classify heart sounds.

Author (Year)	**Meta-heuristic**	**Paper Name**	**Comments**
Franco et al. (2017) [13]	Statistical Algorithms	Impact of prealbumin on mortality and hospital readmission in patients with acute heart failure	Statistical Analysis is used to find the mortality and readmission rates.
Esteva et al. (2017) [12]	Convolution Neural Network	Dermatologist-level classification of skin cancer with deep neural networks	Training of Skin lesions using a single CNN is done on an end-to-end basis from images directly, using only two inputs- pixels and diseases.
Biswal et al. (2017) [3]	SLEEPNET	SLEEPNET: automated sleep staging system via deep learning	Offer SLEEPNET (Sleep EEG neural network), a deployed annotation tool for sleep staging.
Che et al. (2018) [5]	Gated Recurrent Unit	Recurrent neural networks for multivariate time series with missing values	Uses GRU-D to capture long-term temporal dependencies in time series and utilizes the missing patterns to achieve better prediction results.
Reddy and Delen (2018) [29]	LSTM	Predicting hospital readmission for lupus patients: An RNN-LSTM-based deep-learning methodology	Uses deep learning methods to predict rehospitalization within 30 days by extracting the temporal relationships in the longitudinal EHR clinical data.

Author (Year)	Meta-heuristic	Paper Name	Comments
Xiong et al. (2018) [33]	Convolution Recurrent Neural Network	ECG signal classification for the detection of cardiac arrhythmias using a convolutional recurrent neural network	Developed RhythmNet, a 21-layer 1D convolutional recurrent neural network to classify ECGs of different rhythms including AF automatically.
Biswas et al. (2018) [4]	Convolution based encoder & decoder	Deep learning strategy for accurate carotid intima-media thickness measurement: An ultrasound study on Japanese diabetic cohort	The Deep Learning system can be used for stroke risk assessment during routine or clinical trial modes.
Acharya et al. (2018) [1]	Convolution Neural Network	Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals	A Novel method for UCAV path planning in a 2D environment. A 13-layer deep learning CNN algorithm is implemented for the automated EEG analysis.

2.3 Research gaps

In the literature survey, we find the following research gaps. These are points that we will be focused on in this thesis.

- The results obtained in these techniques can be improved using novel classification algorithms. Most of the researchers have used existing machine learning algorithms, and it is expected that deep learning techniques will surpass conventional methods.

- These earlier research were unable to model high dimensional nonlinear relations as good as RNN.
- Descriptive statistics were being used in earlier methods. However, these statistics like mean, median & mode are always under the risk of losing some vital information.
- Deep learning algorithms are still under a shadow for a variety of healthcare applications.

2.4 Novelity

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

2.5 Conclusion

Halth Care Readmissions is extremely difficult thing for both sufferer and health center. In our thesis, we design a robust algorithm to determine the amount of patients readmitted to a health care. We compared 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 machine learning 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 structure can be used in current's health system to aim at high possibility patients, decrease the degree of readmission, and provide excellent health care.

Chapter 3

Methodology

3.1 Data description: MIMIC-III

Medical Information Mart for Intensive Care (MIMIC-III)[23] consists of data about patients enrolled to various critical care units in a large hospital. 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 such as observations, fluid balance, procedure codes, vital signs, medications, laboratory measurements, 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 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 vast 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 of data. Clinical variables are recorded concerning time for each patient. Figure 1 shows the chart of timing clinical variables recorded in the dataset.

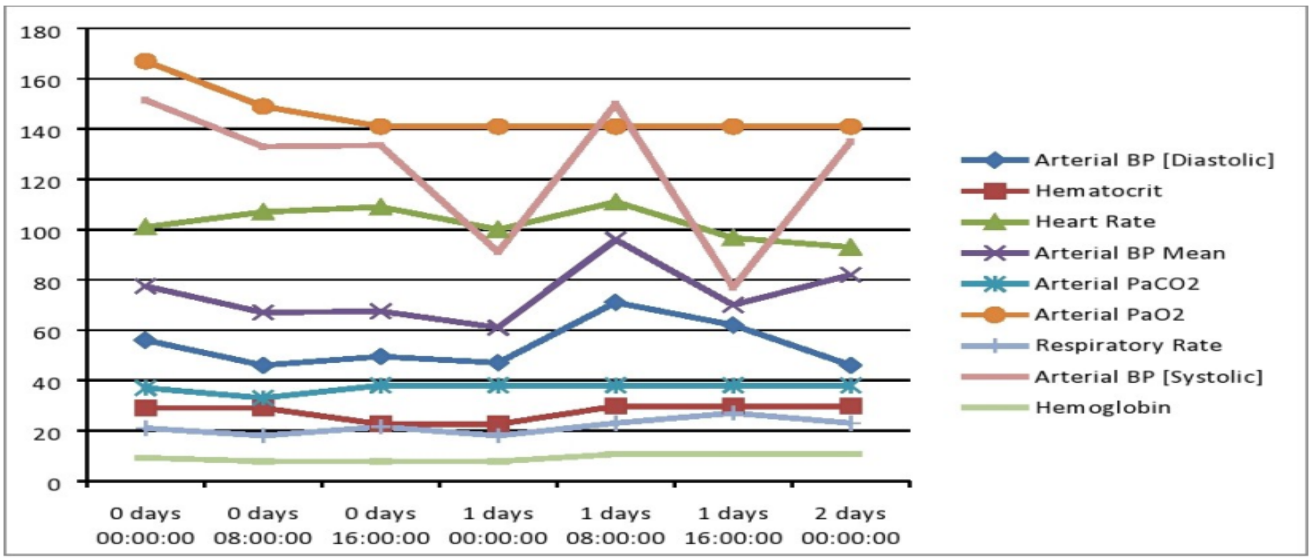

Figure 3.1: Some of the Clinical recordings associated with each patient[23]

3.2 Mechanism/Algorithm

3.2.1 Dataset Preprocessing

3.2.1.1 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 7,534 patients. The data set is divided into two classes:-

- Patients admitted again in 30 days.
- Patients admitted again after 30 days.

3.2.1.2 Missing Values

MIMIC-III dataset contains missing values in some of the features. The feature will be removed if it contains more missing value otherwise mean will be used to fill the missing values.

3.2.1.3 Normalization

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

- Min-Max Scaling

  Let a matrix "M":-

$$M = \begin{Vmatrix} m11 & m12 \\ m21 & m22 \end{Vmatrix} \tag{3.1}$$

  So for scaling we compute

$$m = min(m11, m21), \tag{3.2}$$

$$b = max(m11, m12) \tag{3.3}$$

$$c = min(m12, m22), \tag{3.4}$$

$$d = max(m21, m22) \tag{3.5}$$

  And "M" becomes:-

$$M = \begin{Vmatrix} (m11 - a)/(b - a) & (m12 - c)/(d - c) \\ (m21 - a)/(b - a) & (m22 - c)/(d - c) \end{Vmatrix} \tag{3.6}$$

3.2.2 Feature Selection

Features are selected through Nature Inspired Algorithms:

3.2.2.1 Bat Algorithm

Bat Algorithm is an optimization algorithm that mimics the echolocation behavior of microbats[34]. 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 position x_i with a fixed frequency μ_{min}, varying wavelength and loudness L_0 to look for the prey. They can automatically change the wavelength (or 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 change in many ways, we assume that the loudness changes from a high (positive) L_0 to a minimum constant value L_{min}. Rules to update position x_i and speed u_i of bats at time step t is given by:

$$\mu_i = \mu_{min} + (\mu_{max} - \mu_{min})\beta \tag{3.7}$$

$$u_i^t = u_i^{t-1} + (x_i^{t-1} - x_b)\mu_i \tag{3.8}$$

$$x_i^t = x_i^{t-1} + u_i^t \tag{3.9}$$

where $\beta \in$[0,1] is a arbitrary vector extract from a uniform 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 random walk is used to find a new solution for each bat once a solution is selected from the current best solution:

$$x_{new} = x_{old} + \epsilon L^t \tag{3.10}$$

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

$$L_i^{t+1} = \alpha L_i^t \tag{3.11}$$

$$r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \tag{3.12}$$

where α & γ are constant.

3.2.2.2 Grey Wolf Optimizer

Grey Wolf Optimizer(GWO) Algorithm mimics the behaviour of grey wolves[27]. The GWO algorithm imitates the leadership pecking order and hunting procedure of grey wolves in nature. Four types of grey wolves are selected for replicate the leadership hierarchy such as alpha, beta, delta, and omega. Also, the three main steps of hunting, 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

Searching

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

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

$$\overrightarrow{\mathrm{d}} = |\overrightarrow{\mathrm{c}}\,\overrightarrow{\mathrm{x}}_p(t) - \overrightarrow{\mathrm{x}}(t)| \tag{3.13}$$

$$\overrightarrow{\mathrm{x}}(t+1) = \overrightarrow{\mathrm{x}}_p(t) - \overrightarrow{\mathrm{w}}\,\overrightarrow{\mathrm{d}} \tag{3.14}$$

Where t is the current iteration, $\overrightarrow{\mathrm{w}}$, $\overrightarrow{\mathrm{c}}$ are coefficient vectors and $\overrightarrow{\mathrm{x}}_p$ is position of prey and $\overrightarrow{\mathrm{x}}$ is the position of grey wolf.

Updation of Coefficients

$$\overrightarrow{\mathrm{w}} = 2\overrightarrow{\mathrm{a}}\,\overrightarrow{\mathrm{r1}} - \overrightarrow{\mathrm{a}} \tag{3.15}$$

$$\overrightarrow{\mathrm{c}} = 2\overrightarrow{\mathrm{r2}} \tag{3.16}$$

Where $\overrightarrow{\mathrm{a}}$ linearly decrease from 2 to 0 over the course of iteration. $\overrightarrow{\mathrm{r1}}$ and $\overrightarrow{\mathrm{r2}}$ are arbitrary vectors $\in$ [0,1]. **Attacking** The hunt 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 positions of other agents based on these three. It is modeled as:

$$\overrightarrow{\mathrm{d}}_\alpha = |\overrightarrow{\mathrm{c}}_1\overrightarrow{\mathrm{x}}_\alpha - \overrightarrow{\mathrm{x}}| \tag{3.17}$$

$$\overrightarrow{\mathrm{d}}_\beta = |\overrightarrow{\mathrm{c}}_2\overrightarrow{\mathrm{x}}_\beta - \overrightarrow{\mathrm{x}}| \tag{3.18}$$

$$\overrightarrow{\mathrm{d}}_\delta = |\overrightarrow{\mathrm{c}}_3\overrightarrow{\mathrm{x}}_\delta - \overrightarrow{\mathrm{x}}| \tag{3.19}$$

$$\overrightarrow{\mathrm{x}}_1 = \overrightarrow{\mathrm{x}}_\alpha - \overrightarrow{\mathrm{w}}_1\overrightarrow{\mathrm{d}}_\alpha \tag{3.20}$$

$$\overrightarrow{x}_2 = \overrightarrow{x}_\beta - \overrightarrow{w}_2 \overrightarrow{d}_\beta \tag{3.21}$$

$$\overrightarrow{x}_3 = \overrightarrow{x}_\delta - \overrightarrow{w}_3 \overrightarrow{d}_\delta \tag{3.22}$$

$$\overrightarrow{x}(t+1) = (\overrightarrow{x}_1 + \overrightarrow{x}_2 + \overrightarrow{x}_3)/3 \tag{3.23}$$

Grey wolf attack it's target when it is immobile. In GWO vector w is a arbitrary value within an interval [-2a,2a] a decrease from 2 to 0 throughout the iteration. When $|w| < 1$ wolves attack prey.

3.2.3 Feature Extraction

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

3.2.3.1 Convolutional Neural Network

A convolution neural network(CNN) is an efficient machine learning approach from deep learning and is very similar to standard Neural Networks. It consists of three layers:

- Convolution Layer
- Pooling Layer
- Fully Connected Layer

Convolution Layer 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:

$$WI^x + b \tag{3.24}$$

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 original RGB image, depth is equal to three. Given an image of size (5x5x3) and given ten filters of size (3x3x3). Here, when the stride is equal to 1, the filters runs a single pixel over an image. In this case, an output neuron is of size (3x3x10). Whereas when the stride is equal to 2, the filters will skip two pixels, and output neuron size 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.

$$Output = (W - F + 2P)/S + 1 \tag{3.25}$$

where W is the input neurons size, F is the filter (kernel) size, P is the size of the padding, and S is the stride. Linear algebra are also used in the convolutional neural network. Now consider that given a matrix of size AxB where A is the number of rows and B is the number 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:

$$P(i, j) = \sum_{a=0}^{A_M - 1} \sum_{b=0}^{B_M - 1} M(a, b) * N(i - a, j - b) \tag{3.26}$$

Max-pooling layer: Pooling layer executes the next operation after each convolutional layer. These layers are used to reduce the size of the neurons. These are small rectangular grids that acquires small portion of convolutional layer and filters it to give an output from that block. The most commonly used method is max pooling that fetch that maximum pixel from the block. A formulation for a single type of the pooling layer, max-pooling is presented in equation

$$h_l^j(a, b) = max \in \mathcal{N}(a), b \in \mathcal{N}(b) h_j^{l-1}(\bar{x}, \bar{b}) \tag{3.27}$$

Fully Connected layers: The final layer of a convolutional neural network(cnn) is the 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 from all input neurons through all output neurons.

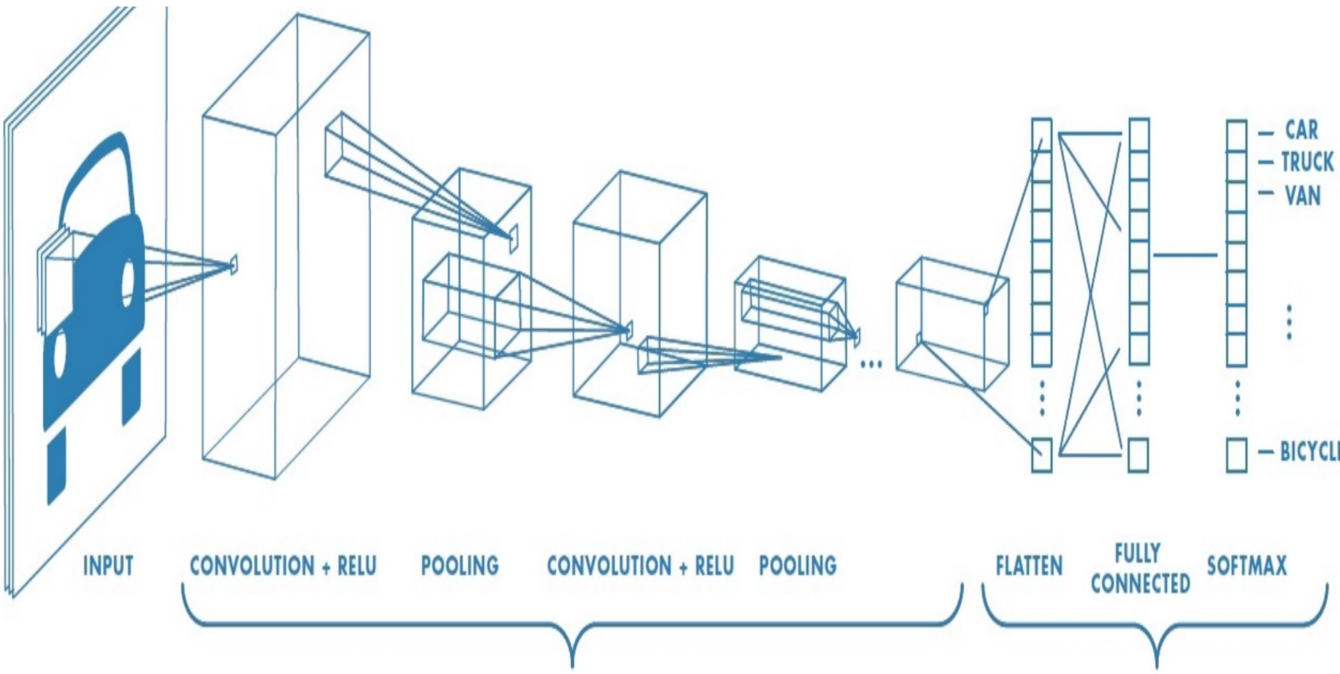

Figure 3.2: Convolution Neural Network Architecture

3.2.4 Algorithms

3.2.4.1 Recurrent Neural Network

Recurrent neural networks (RNN) are used to represent the temporal dependency in the time series data[5]. 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:

$$y_i^t = \sigma(W_i x^t + b_i) \tag{3.28}$$

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

$$y_i^t = (W_i x^t + V_i x^{t-1} + b_i) \tag{3.29}$$

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 issue of vanishing gradients and were developed to model long-range dependencies efficiently. LSTMs can fix this by maintaining an internal state that represents 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.

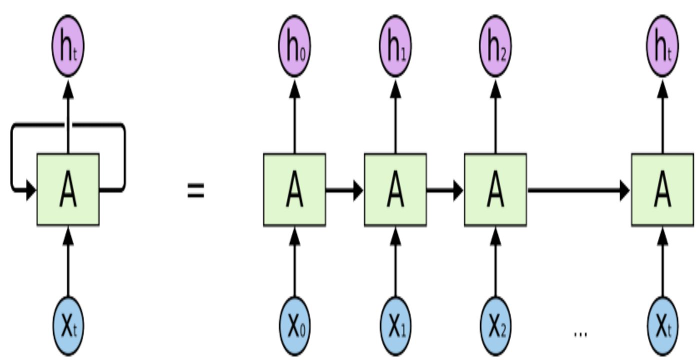

Figure 3.3: An unrolled recurrent neural network[17]

3.2.4.2 Long Short Term Memory(LSTM) Networks

Long Short Term Memory networks are usually known as LSTMs are a particular kind of RNN, that are good in understanding long-term dependencies. They were introduced by [19] 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 modules of the neural network. In traditional RNNs, this replicating module have a straightforward architecture, such as a single tank layer. LSTMs also have this series-like architecture, but the recurrent module has a different architecture. Rather than having a single neural network 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.

$$f_t = \sigma(W_f X_t + R_f l_{t-1} + b_f) \tag{3.30}$$

$$i_t = \sigma(W_i X_t + R_i l_{t-1} + b_i) \tag{3.31}$$

$$o_t = \sigma(W_o X_t + R_o l_{t-1} + b_o) \tag{3.32}$$

$$\tilde{C}_t = \Phi(W_C X_t + R_C l_{t-1} + b_c) \tag{3.33}$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \tag{3.34}$$

$$l_t = o_t \phi(C_t) \tag{3.35}$$

i_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 remove or add information 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, i, 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 forgetting the things we decided to forget. Then we add i_t*δC_t in it. This is the new candidate values, increased by how much we decided to update each state value. The final output of an LSTM unit is given by equation(3.35). Here * represents the Hadamard (element-wise) multiplication operation. Figure 3 shows the LSTM network.

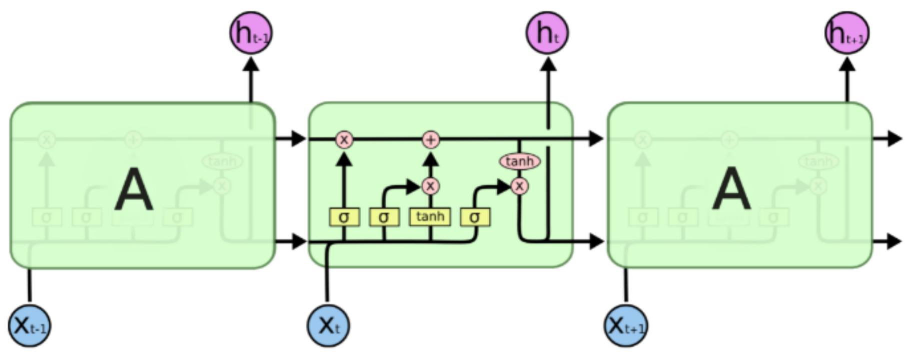

Figure 3.4: The replicating module in an LSTM[17]

3.2.4.3 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[31]. This algorithm gives the better result in both regression and classification problem. BRNN computes both forward ($\overrightarrow{l}$) and backward ($\overleftarrow{l}$) hidden sequence.

$$\overrightarrow{l}_t = \mathcal{O}(W_{x\overrightarrow{l}} x_t + W_{\overrightarrow{l}\overrightarrow{l}} \overrightarrow{l}_{t-1} + b_{\overrightarrow{l}}) \tag{3.36}$$

$$\overleftarrow{\rceil}_t = \mathcal{O}(W_{x\overleftarrow{\rceil}} x_t + W_{\overleftarrow{\rceil}\overleftarrow{\rceil}} \overleftarrow{\rceil}_{t+1} + b_{\overleftarrow{\rceil}}) \tag{3.37}$$

$$y_t = W_{\overrightarrow{\rceil}y} \overrightarrow{\rceil}_t + W_{\overleftarrow{\rceil}y} \overleftarrow{\rceil}_t + b_o \tag{3.38}$$

The long range context can be accessed in both directions by combining BRNNs with LSTM which gives bidirectional LSTM[16].

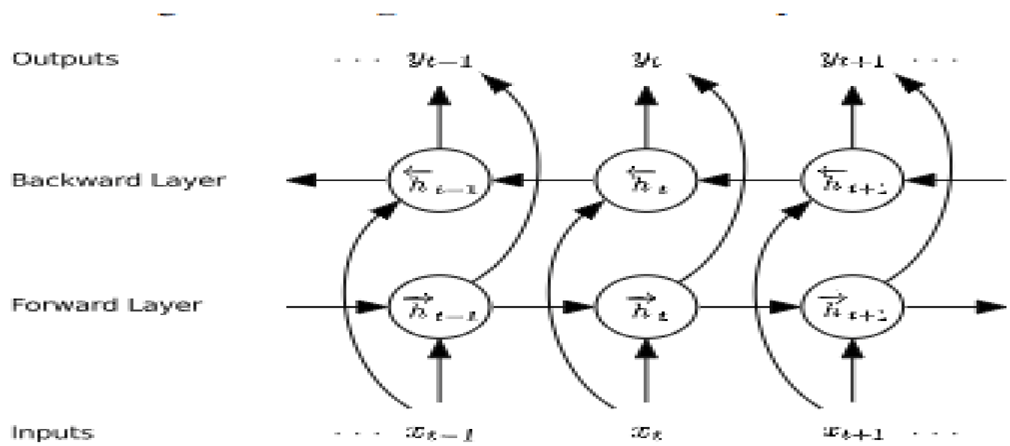

Figure 3.5: Bidirectional Recurrent Neural Networks[15]

3.2.4.4 Auto Encoders

An auto-encoder[26] 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.

3.3 Conclusion

The methodology and the algorithms used to compute, generate, test, and verify the process produces good results.

Chapter 4

Experiments and results

This section discusses the implementation details and the results obtained using varoius algorithms.

4.1 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 long short term memory(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 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.

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

4.2 Results Obtained

This segment shows the readmission prediction results acquired from various predictive models. The prediction accuracy will be differentiate on the basis of two variables AUC under ROC (Receiver Operating Characteristic) Curve[22], and final accuracy. Both area under ROC curve and final accuracy relies on the classifiers capability to rank examples for positive class, however in the case of final accuracy, it also relies on the capability to compute the threshold in the ordering to distinguish the positive class. Applying various models on the dataset, with default attributes, we got results as referenced in Table 1.

Algorithm type	**Algorithm**	**AUC**	**Accuracy**	**F1 Score Micro**	**F1 Score Macro**
Deep Learning without Nature Inspired Algorithms	Long Short Term Memory	0.88	81.71%	0.80	0.76
	Long Short Term Memory Auto-Encoders	0.73	75.00%	0.70	0.62
	Convolution Long Short Term Memory	0.89	85.63%	0.71	0.66
	Convolution Long Short Term Memory Auto-Encoders	0.87	82.41%	0.80	0.78
	Bi-Directional Long Short Term Memory Auto-Encoders	0.85	83.42%	0.78	0.75
Deep Learning with Nature Inspired Algorithms	Long Short Term Memory with Bat Algorithm	0.75	77.58%	0.70	0.63
	Long Short Term Memory with Grey Wolf optimizer Algorithm	0.85	87.48%	0.79	0.76

Table 4.1: Comparison among different Models implemented so far

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 4.2: Comparison with Baseline Models

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

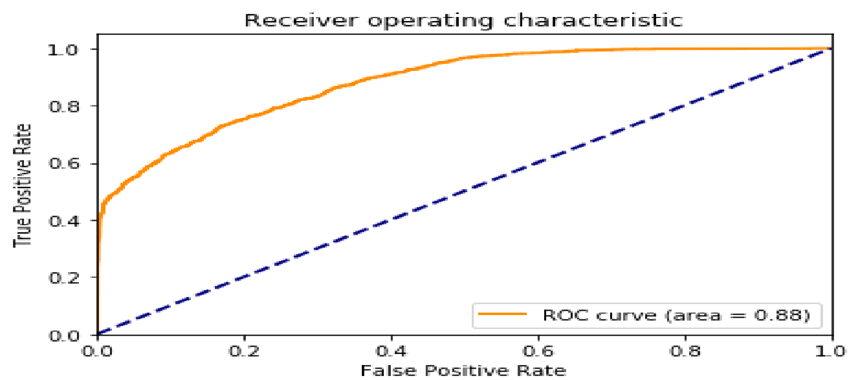

Figure 4.1: ROC Curve: Long Short Term Memory.

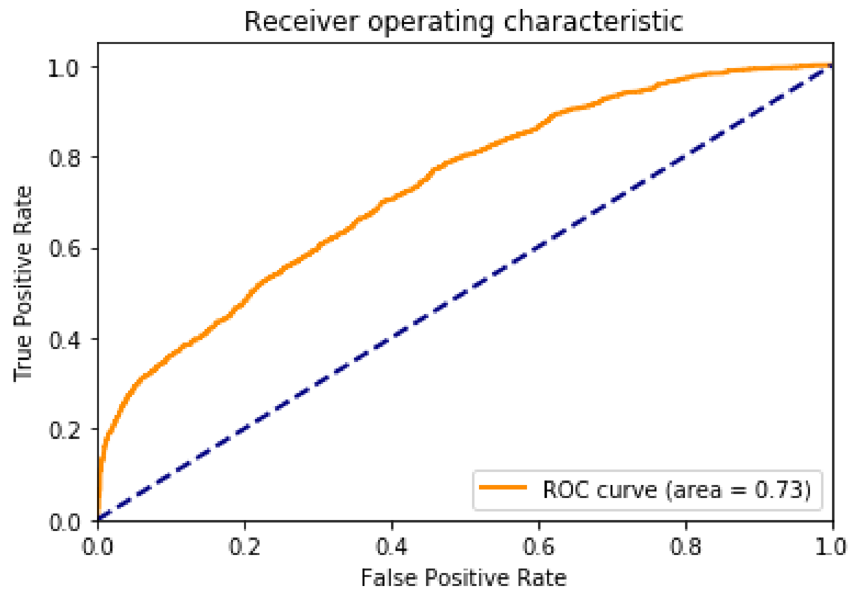

Figure 4.2: ROC Curve: Long Short Term Memory Auto-Encoders.

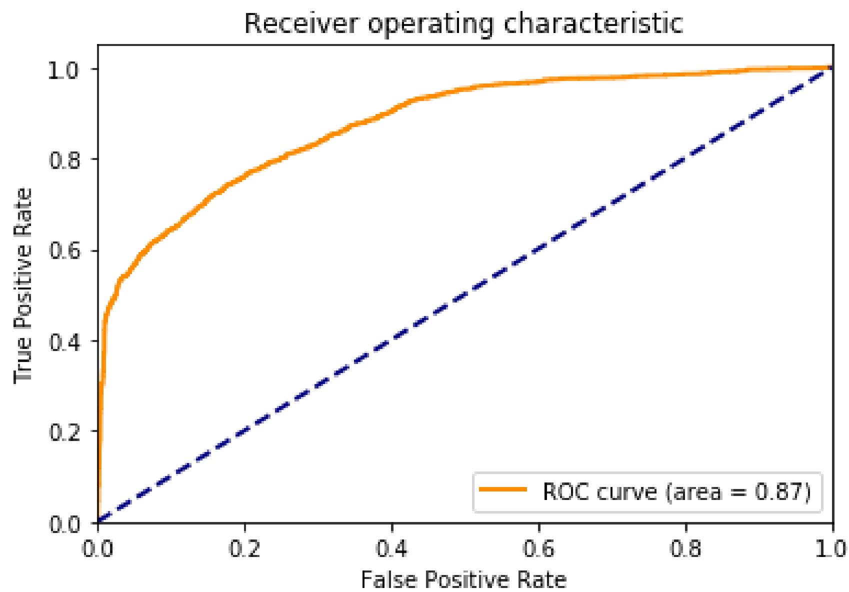

Figure 4.3: ROC Curve: Convolution Long Short Term Memory.

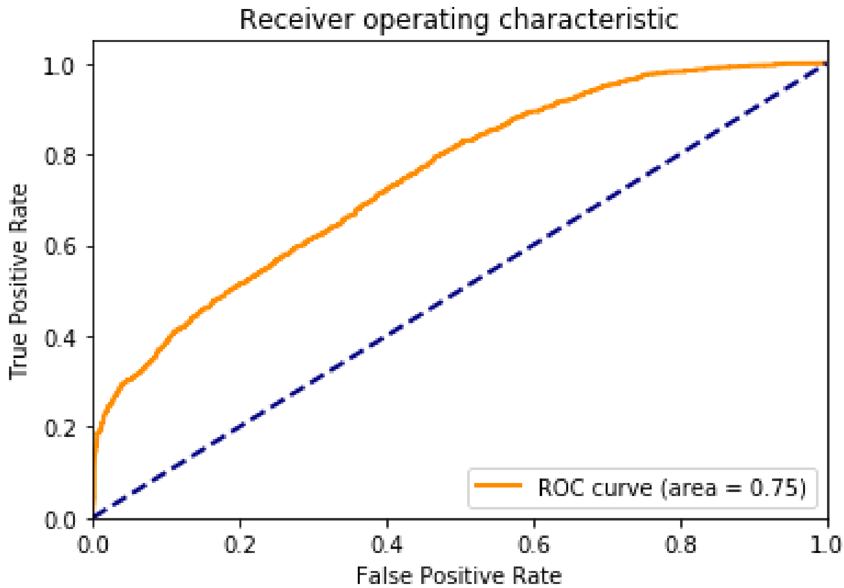

Figure 4.4: ROC Curve: Long Short Term Memory with Bat Algorithm.

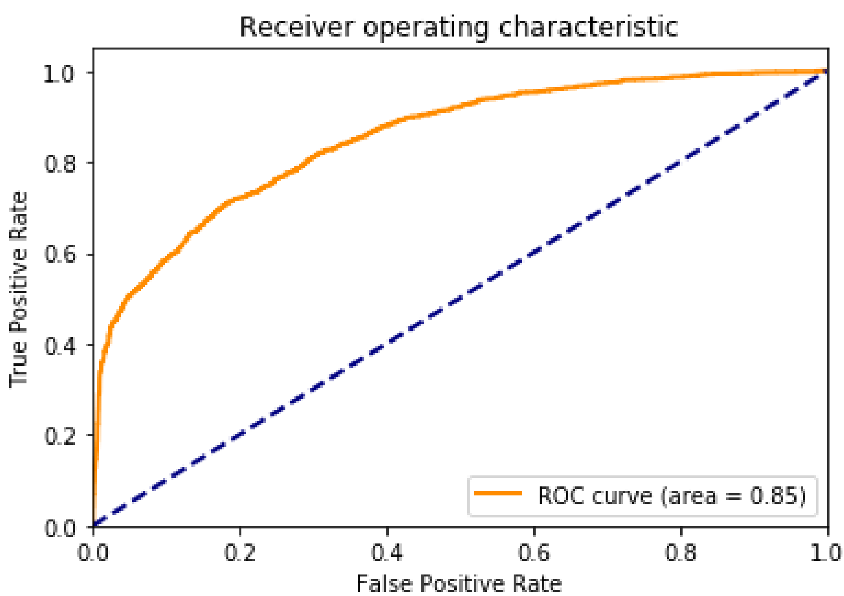

Figure 4.5: ROC Curve: Long Short Term Memory with Grey Wolf Optimizer Algorithm.

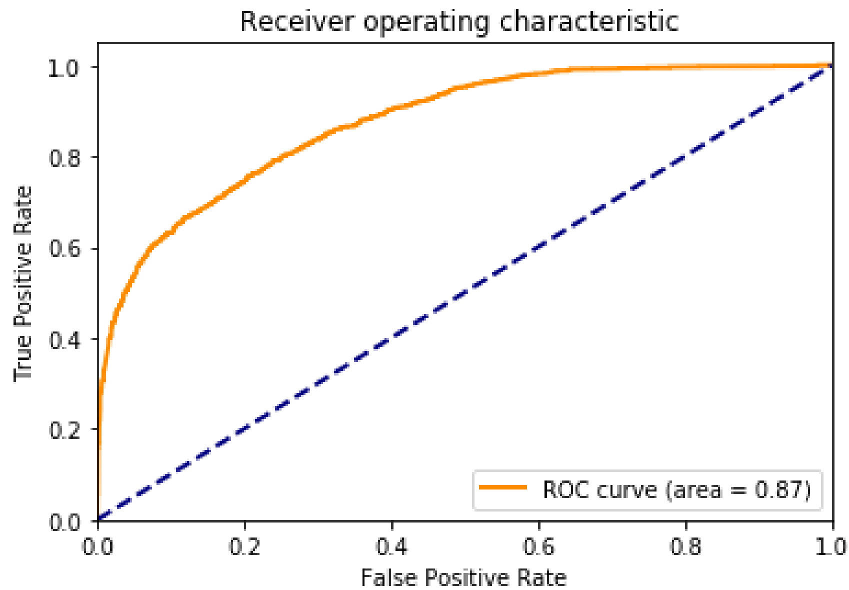

Figure 4.6: ROC Curve Convolution Long Short Term Memory Auto Encoders.

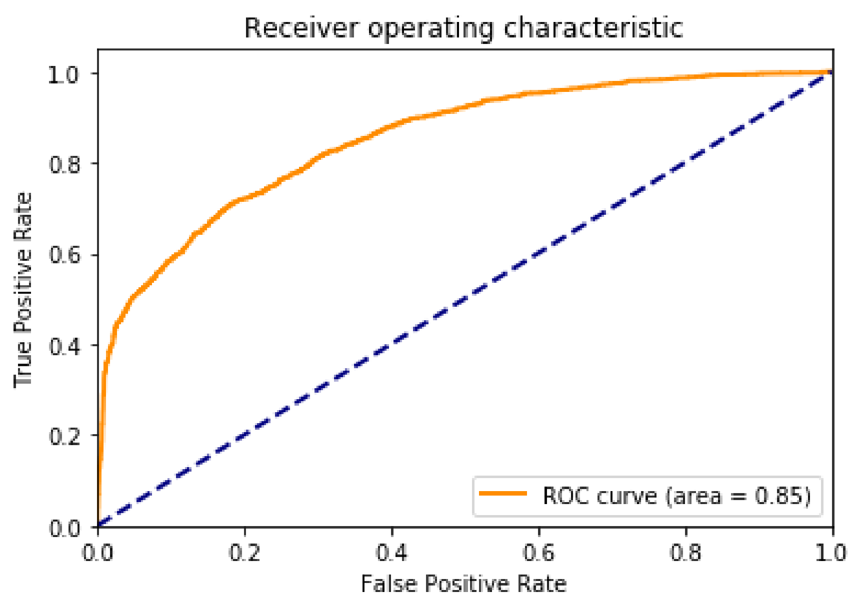

Figure 4.7: ROC Curve Bi-Directional Long Short Term Memory.

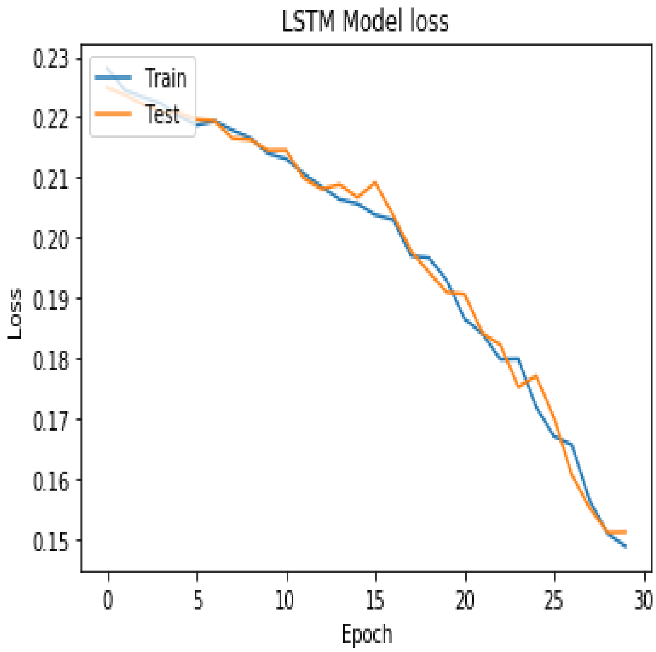

Figure 4.8: Loss Curve: Long Short Term Memory.

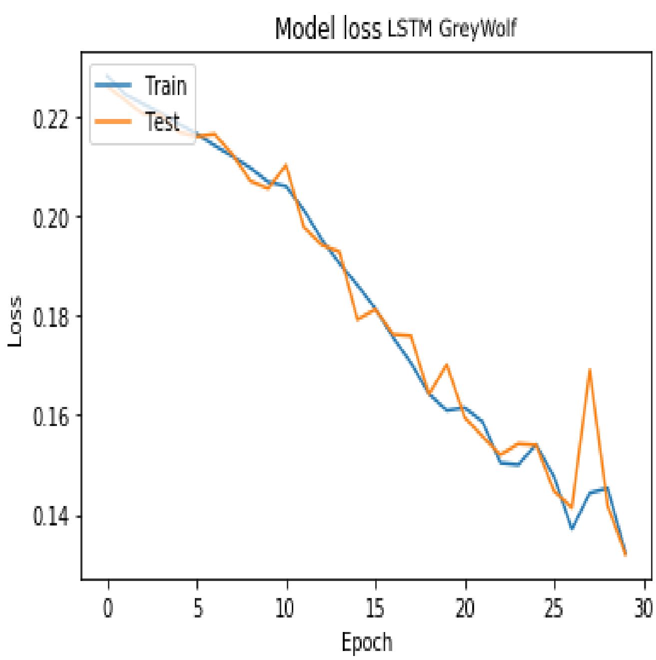

Figure 4.9: Loss Curve: Long Short Term Memory Grey Wolf.

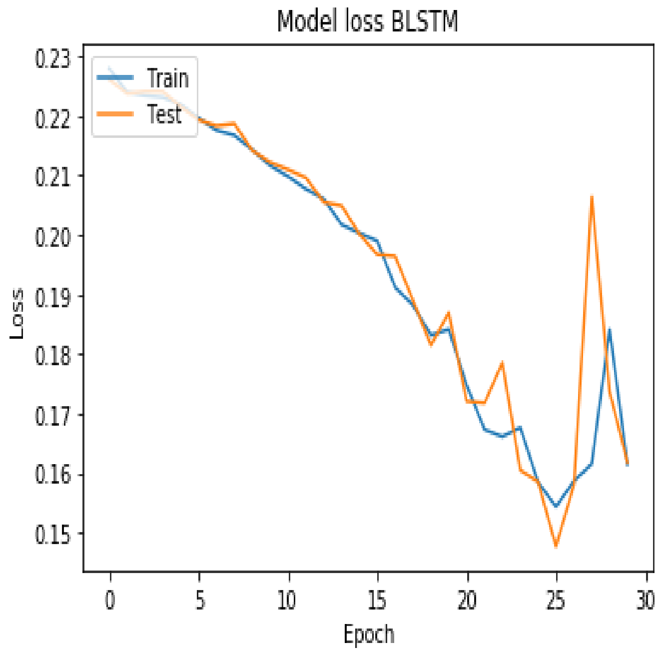

Figure 4.10: Loss Curve: Bidirectional Long Short Term Memory.

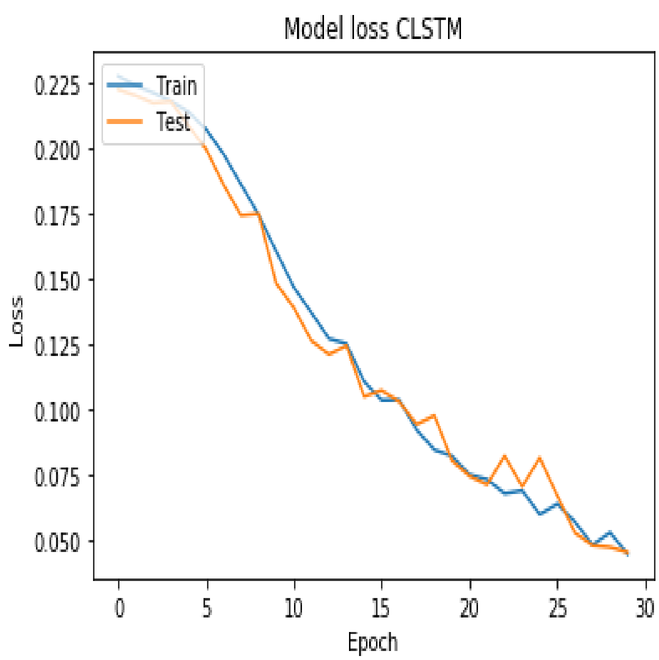

Figure 4.11: Loss Curve: Convolution Long Short Term Memory.

Chapter 5

Discussions and conclusion

5.1 Contributions

Following are the contributions made in the research work.

- To compare the performance of various deep learning algorithms on MIMIC-III dataset based on area under the roc curve, accuracy anf F1Scores.
- Compare the results of deep learning models with baseline models to point the advantages of deep learning in the field of health-care applications.
- Comparison of deep learning models with or without nature-inspired algorithms.

5.2 Limitations

The computation power and the memory required to process and compute such a big dataset, and the heavy amounts of output becomes one of the major limitations of the model. Since the data is already 35gb and as well as we use the LSTM model and hybrid model, i.e., CLSTM also with nature-inspired algorithms which store a lot of data while computation, therefore memory requirements becomes considerably large.

5.3 Future scope

Presently, this study only targets patients who were admitted again to the hospital for sure, whether within 30 days or after 30 days. This study can be extended, and one can classify the data in three categories:

- Patients admitted again in 30 days.
- Patients admitted again after 30 days.
- Patients that were never readmitted.

Bibliography

[1] U Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Hojjat Adeli. Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals. Computers in biology and medicine, 100:270–278, 2018.

[2] Yoshua Bengio et al. Learning deep architectures for ai. Foundations and trends® in Machine Learning, 2(1):1–127, 2009.

[3] Siddharth Biswal, Joshua A Kulas, Haoqi Sun, Balaji Goparaju, M. Brandon Westover, Matt T. Bianchi, and Jimeng Sun. Sleepnet: Automated sleep staging system via deep learning. CoRR, abs/1707.08262, 2017.

[4] Mainak Biswas, Venkatanareshbabu Kuppili, Tadashi Araki, Damodar Reddy Edla, Elisa Cuadrado Godia, Luca Saba, Harman S Suri, Tomaž Omerzu, John R Laird, Narendra N Khanna, et al. Deep learning strategy for accurate carotid intima-media thickness measurement: An ultrasound study on japanese diabetic cohort. Computers in biology and medicine, 98:100–117, 2018.

[5] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. Scientific reports, 8(1):6085, 2018.

[6] 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.

[7] Zhengping Che, Sanjay Purushotham, Robinder G. Khemani, and Yan Liu. Distilling knowledge from deep networks with applications to healthcare domain. CoRR, abs/1512.03542, 2015.

[8] Zhengping Che, Jennifer St Sauver, Hongfang Liu, and Yan Liu. Deep learning solutions for classifying patients on opioid use. In AMIA Annual Symposium Proceedings, volume 2017, page 525. American Medical Informatics Association, 2017.

[9] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In Machine Learning for Healthcare Conference, pages 301–318, 2016.

[10] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. Journal of the American Medical Informatics Association, 24(2):361–370, 2016.

[11] Rahul Dey and Fathi M Salemt. Gate-variants of gated recurrent unit (gru) neural networks. In Circuits and Systems (MWSCAS), 2017 IEEE 60th International Midwest Symposium on, pages 1597–1600. IEEE, 2017.

[12] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639):115, 2017.

[13] 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.

[14] 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.

[15] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In International Conference on Machine Learning, pages 1764–1772, 2014.

[16] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks, 18(5-6):602–610, 2005.

[17] 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.

[18] 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.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, pages 448–456. JMLR.org, 2015.

[21] Y Kumar Jain and Santosh Kumar Bhandare. Min max normalization based data perturbation method for privacy protection. International Journal of Computer & Communication Technology, 2(8):45–50, 2011.

[22] Jonathan E Fieldsend JE. Formulation and comparison of multi-class roc surfaces. 2005.

[23] 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.

[24] 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.

[25] Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. CoRR, abs/1506.00019, 2015.

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

[27] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. Advances in engineering software, 69:46–61, 2014.

[28] 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 (sicula): a population-based study. The Lancet Respiratory Medicine, 3(1):42–52, 2015.

[29] 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.

[30] Jonathan Rubin, Rui Abreu, Anurag Ganguli, Saigopal Nelaturi, Ion Matei, and Kumar Sricharan. Recognizing abnormal heart sounds using deep learning. In KHD@IJCAI, 2017.

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

[32] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. BioMed research international, 2014, 2014.

[33] Zhaohan Xiong, Martyn P Nash, Elizabeth Cheng, Vadim V Fedorov, Martin K Stiles, and Jichao Zhao. Ecg signal classification for the detection of cardiac arrhythmias using a convolutional recurrent neural network. Physiological measurement, 39(9):094006, 2018.

[34] Xin-She Yang and Amir Hossein Gandomi. Bat algorithm: a novel approach for global engineering optimization. Engineering Computations, 29(5):464–483, 2012.