International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT217261

Implementation of Healthcare Aid for Ailment Detection and Remedy using NLP and Full Stack

Gaurav Kumar Daharia[1], Mayank Tiwari[1], Priyanka Gupta[2]
[1]Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India
[2]School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India

ABSTRACT

Article Info
Volume 7, Issue 2
Page Number: 317-323
Publication Issue :
March-April-2021
Article History
Accepted : 12 April 2021
Published : 17 April 2021

Diseases are one of the challenges that the human race has ever faced, humans have developed new things every time to get a cure for it, but the requirements change as time changes, so through this general, we are trying to provide help to upcoming generation with the help of some advance tools.

Keywords : Natural Language Processing, Data Preprocessing, Full stack, deep learning

I. INTRODUCTION

In this world of rapid changes everyone wants to be quick and fast, nobody wants to wait for a penny of a second, but people don't know about what kind of adverse effect this kind of huge pace life can cost. This kind of fast life can cause different kinds of health problems, those can be general or severe once. So, from this paper, we are trying to provide benefits to health care fields with help of Natural Language Processing and in collaboration with Full stack, so that we can use these techs to get a boost in disease detection from the symptoms provided by the users.

II. Literature Review

This section reviews literature used in this paper.

Heart Disease: Heart-related diseases also known as Coronary Heart Diseases (CHD), which is the problem of deposition of fats inside the blood tubes passing the blood to the heart muscles. Heart diseases could occur as early as 18 years and they could be detected when the blockage exceeds about 70%. If these blockages remain undetected or not treated then could cause rupturing of the membrane covering the blockage because of the excess pressure of blood flow. The mixing of the chemicals released from the membrane with the blood could lead to a blood clot and would excessively lead to various Heart diseases [7].

The reasons which increase blockage are called risk factors. These risk factors are classified as modifiable and non-modifiable risk factors. Non-modifiable risk factors are age, gender, and heredity. These risk factors cannot be modified, and they will always keep causing heart disease. Risk factors that can be changed by our efforts are called modifiable risk factors. Some modifiable risk factors are 1) Food-related 2) Habit related 3) Stress-related 4) Biochemical and miscellaneous risk factors.

Atherosclerosis, coronary, congenital, rheumatic, myocarditis, arrhythmia, and angina are the different types of heart diseases [8]. Common symptoms of heart disease are listed in Table 1[9].

Table 1: Symptoms of heart disease [9].

Sl.no	Symptoms name
1.	Chest pain
2.	Strong compressing or flaming in the chest
3.	Discomfort in the chest area
4.	Sweating
5.	Lightheadedness
6.	Dizziness
7.	Shortness of breath
8.	Pain spanning from the chest to arm and neck
9.	Cough
10.	Fluid retention

Table 2 : Risk factors of heart disease [10].

1.	Diabetes
2.	High blood pressure
3.	High LDL
4.	Low HDL
5.	Not getting enough physical activity
6.	Obesity

III. Methodology

1. Recurrent Neural Networks

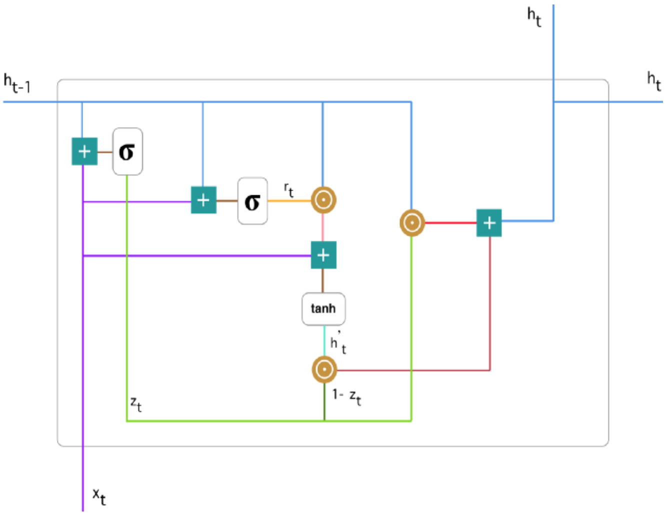
We have already mentioned Natural Language Processing, so here we are sharing which type of network we have used (i.e., GRU or LSTM).

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable-length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition, or speech recognition. [11]

From the above definition, we can see that **RNN's** are very efficient with text data especially when we talk about sequential data where input sequence matters. So, for developing the project we have Gated Recurrent Units (**GRU**) which is more efficient for NLP.

2. More on Gated recurrent units

Gated recurrent units (**GRU**s) are a gating mechanism in recurrent neural networks, introduced in 2014 by Kyunghyun Cho. The GRU is like a long short-term memory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate. GRU's performance on certain tasks of polyphonic music modeling, speech signal modeling, and natural language processing was found to be like that of LSTM. GRUs have been shown to exhibit better performance on certain smaller and less frequent datasets. [12]

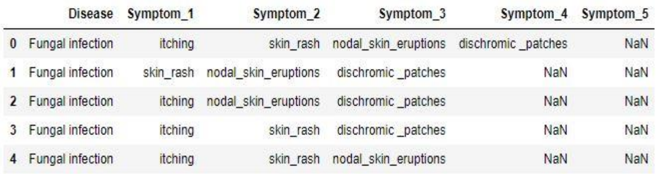


IV. Implementation

Since we have used two tech-stack so first we are going to share how we have processed the text data and how later we have used our trained model with full-stack.

1. Data Gathering

Data collection is the single most important step in solving any problem with help of machine learning. However, it is also a critical roadblock for many researchers and data scientists. An inordinate amount of time is usually spent on data collection, which largely consists of data acquisition, data labeling, and improvement of existing data or models. Teams that dive headfirst into projects without considering the right data collection process often do not get the results they want. We have gathered the data from an open-source platform named **Kaggle** and this dataset is all about diseases and the symptoms that are faced by the patients. Below we have shared a small glimpse of the data.



	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	NaN	NaN
3	Fungal infection	itching	skin_rash	dischromic _patches	NaN	NaN
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	NaN	NaN

From the above glimpse, we can see that the dataset contains some NAN or NULL values so for this we to do some data cleaning and preprocessing so that we can train the model.

2. Data Cleaning and Pre-processing

Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. Data Cleaning is one of those things that everyone does, but no one talks about. It surely is not the fanciest part of machine learning and at the same time, there are not any hidden tricks or secrets to uncover. However, proper data cleaning can make or break your project

So as per the above definition, it is important to do data cleaning, since here we are working upon text data so for that we have done some following steps that are mentioned below.

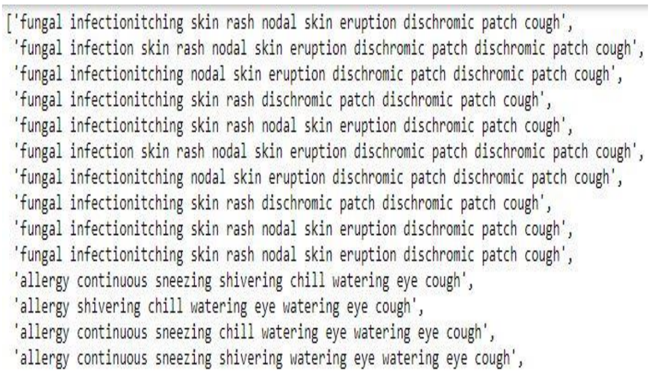
a. Removing all special characters.

b. After removal, we have lowered the text.

c. Then we have created tokens or can say create an array so that we can remove stop words from the text (stop words like a, an, the)

Now by following the above steps, we have cleared the dataset, and, in this way, we are done with our essential step.

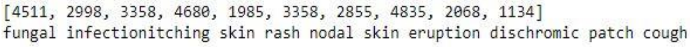
Let us have a small glimpse from our created dataset after following the above steps.



```
['fungal infectionitching skin rash nodal skin eruption dischromic patch cough',
 'fungal infection skin rash nodal skin eruption dischromic patch dischromic patch cough',
 'fungal infectionitching nodal skin eruption dischromic patch dischromic patch cough',
 'fungal infectionitching skin rash dischromic patch dischromic patch cough',
 'fungal infectionitching skin rash nodal skin eruption dischromic patch cough',
 'fungal infection skin rash nodal skin eruption dischromic patch dischromic patch cough',
 'fungal infectionitching nodal skin eruption dischromic patch dischromic patch cough',
 'fungal infectionitching skin rash dischromic patch dischromic patch cough',
 'fungal infectionitching skin rash nodal skin eruption dischromic patch cough',
 'fungal infectionitching skin rash nodal skin eruption dischromic patch cough',
 'allergy continuous sneezing shivering chill watering eye cough',
 'allergy shivering chill watering eye watering eye cough',
 'allergy continuous sneezing chill watering eye watering eye cough',
 'allergy continuous sneezing shivering watering eye watering eye cough',
```

Now we will pre-process the above data in the format that is required for training the model and for that we need to perform the below steps so that we can pre-process data for model training.

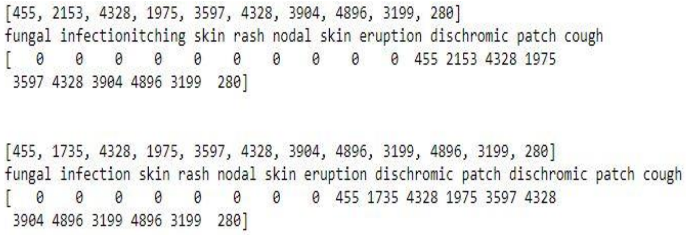
2.1 Creating one hot representation: In this step, we will encode the text into a certain range of number so that they we can train our model. Since our model does not understand the text but it understands numbers so for that reason, we are doing this step.



```
[4511, 2998, 3358, 4680, 1985, 3358, 2855, 4835, 2068, 1134]
fungal infectionitching skin rash nodal skin eruption dischromic patch cough
```

As we can see from here the above pic is the list of symptoms that are encoded in a certain range which is adding some meaning to it.

2.1 Adding Pad to Sequence: We need to add some padding to the sequences because everyone is having a different symptom length so to make all sequences of equal length, we are adding this padding.



```
[455, 2153, 4328, 1975, 3597, 4328, 3904, 4896, 3199, 280]
fungal infectionitching skin rash nodal skin eruption dischromic patch cough
[   0    0    0    0    0    0    0    0    0    0  455 2153 4328 1975
 3597 4328 3904 4896 3199  280]

[455, 1735, 4328, 1975, 3597, 4328, 3904, 4896, 3199, 4896, 3199, 280]
fungal infection skin rash nodal skin eruption dischromic patch dischromic patch cough
[   0    0    0    0    0    0    0    0  455 1735 4328 1975 3597 4328
 3904 4896 3199 4896 3199  280]
```

so in this way, we can prepare as well as pre-process data for model training.

3. Visualization through Full Stack and related technology

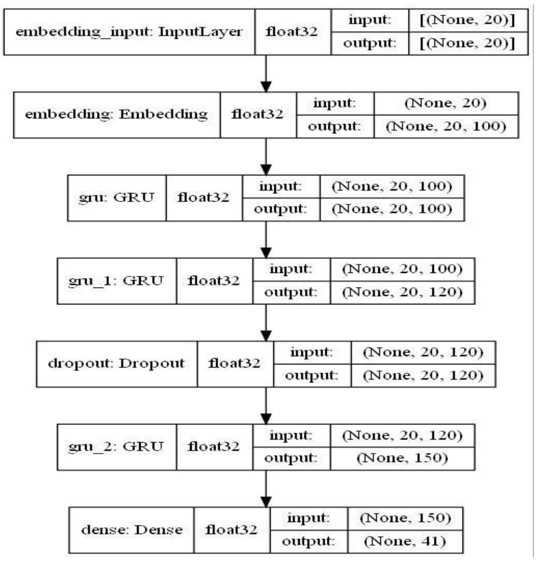
The NLP model is implemented in a website that we have named ‘NOVA HEALTHCARE’. It is made using HTML, CSS, jQuery, MongoDB, etc. The website focuses on helping people understand whether their ailment can be cured at home or a doctor’s visit is required and if a visit is required, provide a channel to access doctors as well.

The website has different pages for all of these scenarios and provides an easy UI for visitors to use.

NOVA HEALTHCARE

Model structure

Now it let’s see more about model structure since it is important to describe more about it because it plays a vital positive role in training the model. After all, layers of the model define the structure of the model, and having good knowledge about each layer will help to create a well-functioning model.


embedding_input: InputLayer | float32 | input: [(None, 20)] | output: [(None, 20)]
embedding: Embedding | float32 | input: (None, 20) | output: (None, 20, 100)
gru: GRU | float32 | input: (None, 20, 100) | output: (None, 20, 100)
gru_1: GRU | float32 | input: (None, 20, 100) | output: (None, 20, 120)
dropout: Dropout | float32 | input: (None, 20, 120) | output: (None, 20, 120)
gru_2: GRU | float32 | input: (None, 20, 120) | output: (None, 150)
dense: Dense | float32 | input: (None, 150) | output: (None, 41)

This model consists of layers like **embedding layer**, **GRU**, **dropout layer**, **dense layer**. These are some layers that we have used to prepare the structure of the model and below we have added a small detail about these layers.

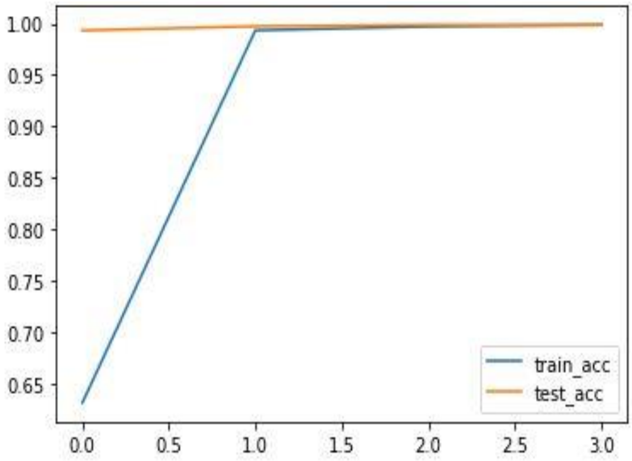
1. **Embedding layer:** This layer is helping us to encode each entry done by the user into a certain dimension which is helping our model to map each entry to its corresponding diseases, in this way our model is doing a part of the prediction.
2. **Dense layer:** This is also a type of layer which allows our model to do some computation.
3. **Dropout layer:** This layer is an especially important layer after the embedding layer, its work is to turn off some nodes so that our model can be able to learn each parameter efficiently and can perform accurate prediction in real-time.

V. RESULTS

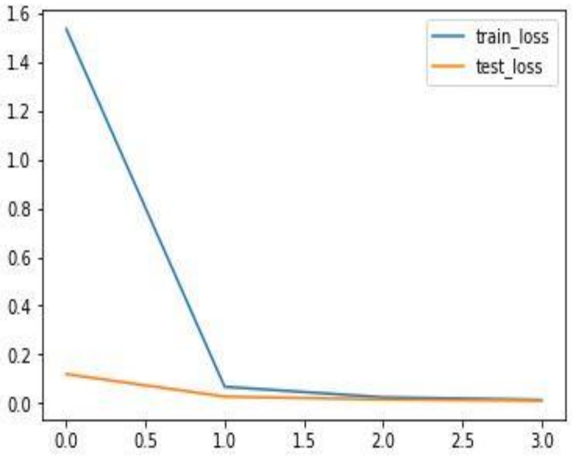
In this section, we are sharing the analysis of model performance because it is also important to see the performance of the model when it is tested on unseen data. Below we have shared two plots that are related to model performance.

1. Model Accuracy Plot

This plot is related to model accuracy where in x-axis is representing the number of iterations, we have used to train the model and the y-axis is representing the accuracy achieved on train data and test data. The shape of whole data after preprocessing is **(4920, 20),** where we have used **(3444, 20)** data for train and **(1476, 20)** for testing. As a part of the performance on training, the model had achieved an accuracy of **99.85%** and on test data, it had achieved an accuracy of **99.88%** which is quite good on unseen data.



2. Model Loss Plot



The above plot is related to the model loss (**error**) wherein the x-axis is representing the number of iterations that we have used to train the model and the y-axis is representing the loss done by the model on train data and test data. As a part of the performance on training, the model had achieved a loss of **1.36%** and on test data, it had achieved a loss of **1.07 %** which is quite good on unseen data. This loss plot is telling us how much percentage of error model is doing on the train as well as on test data and we can see that loss percentage is low on test data as compared to train data this indicates that the model is performing very well.

3. Model Evaluation Parameters

We have certain metrics through which we can judge a model whether is it doing its job correctly or not. For model evaluation we have used certain parameters like **precision, recall** which tells us about how our model is performing, and below we have a small definition about it and a small glimpse of model metrics.

3.1 Precision: Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives. False positives are casing the model incorrectly labels as positive that are negative.[13]

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

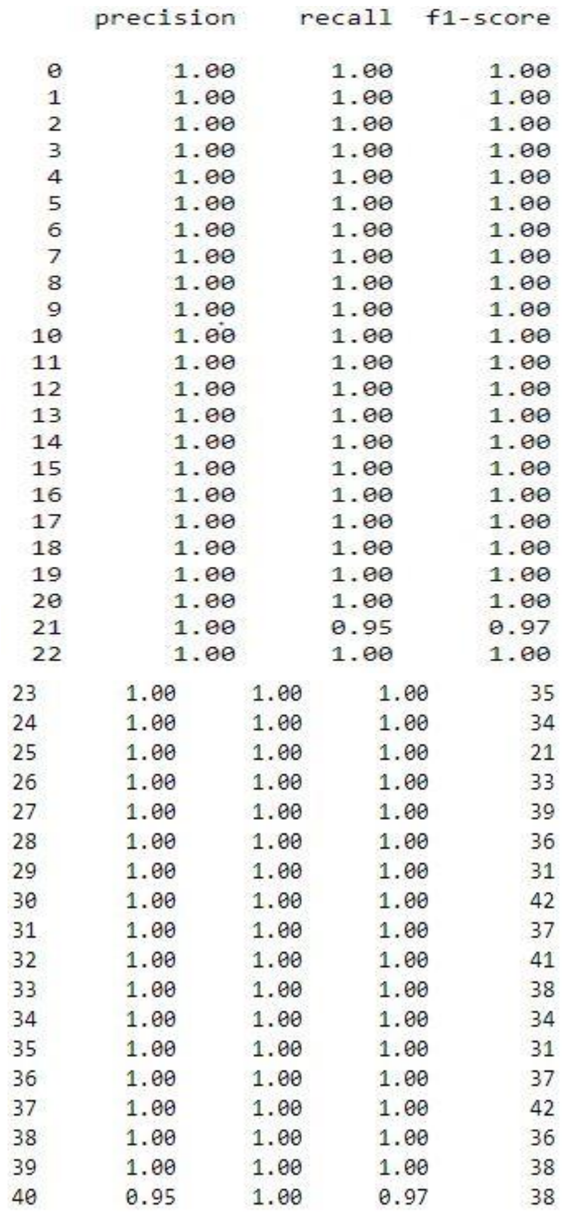
3.2 Recall: Recall quantifies the number of positive class predictions made from all positive examples in the dataset. [13]

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

3.3 Classification Report of Model

This Report tells us about how much our model is working efficiently on test data and this report will also show about precision and recall of the mode Below image the number 0 to 40 is representing the following disease names:

0.'Drug Reaction',
1. 'Malaria',
2. 'Allergy',
3. 'Hypothyroidism',
4. 'Psoriasis',
5. 'GERD',
6. 'Chronic cholestasis',
7. 'hepatitis A',
8. 'Osteoarthristis',
9. '(vertigo) Paroymsal Positional Vertigo',
10. 'Hypoglycemia',
11. 'Acne',
12. 'Diabetes',
13. 'Impetigo',
14. 'Hypertension',
15. 'Peptic ulcer diseae',
16. 'Dimorphic hemorrhoids(piles)',
17. 'Common Cold',
18. 'Chicken pox',
19. 'Cervical spondylosis',
20. 'Hyperthyroidism',
21. 'Urinary tract infection',
22. 'Varicose veins',
23. 'AIDS',
24. 'Paralysis (brain hemorrhage)',
25. 'Typhoid',
26. 'Hepatitis B',
27. 'Fungal infection',
28. 'Hepatitis C',
29. 'Migraine',
30. 'Bronchial Asthma',
31. 'Alcoholic hepatitis',
32. 'Jaundice',
33. 'Hepatitis E',
34. 'Dengue',
35. 'Hepatitis D',
36. 'Heart attack',
37. 'Pneumonia',
38. 'Arthritis',
39. 'Gastroenteritis',
40. 'Tuberculosis'



```
       precision    recall  f1-score

   0        1.00      1.00      1.00
   1        1.00      1.00      1.00
   2        1.00      1.00      1.00
   3        1.00      1.00      1.00
   4        1.00      1.00      1.00
   5        1.00      1.00      1.00
   6        1.00      1.00      1.00
   7        1.00      1.00      1.00
   8        1.00      1.00      1.00
   9        1.00      1.00      1.00
  10        1.00      1.00      1.00
  11        1.00      1.00      1.00
  12        1.00      1.00      1.00
  13        1.00      1.00      1.00
  14        1.00      1.00      1.00
  15        1.00      1.00      1.00
  16        1.00      1.00      1.00
  17        1.00      1.00      1.00
  18        1.00      1.00      1.00
  19        1.00      1.00      1.00
  20        1.00      1.00      1.00
  21        1.00      0.95      0.97
  22        1.00      1.00      1.00
23        1.00      1.00      1.00        35
24        1.00      1.00      1.00        34
25        1.00      1.00      1.00        21
26        1.00      1.00      1.00        33
27        1.00      1.00      1.00        39
28        1.00      1.00      1.00        36
29        1.00      1.00      1.00        31
30        1.00      1.00      1.00        42
31        1.00      1.00      1.00        37
32        1.00      1.00      1.00        41
33        1.00      1.00      1.00        38
34        1.00      1.00      1.00        34
35        1.00      1.00      1.00        31
36        1.00      1.00      1.00        37
37        1.00      1.00      1.00        42
38        1.00      1.00      1.00        36
39        1.00      1.00      1.00        38
40        0.95      1.00      0.97        38
```

VI. REFERENCES

[1]. Onisko A, Druzdzel M.J and Wasyluk H, A Bayesian Network Model for Diagnosis of Liver Disorders. In Proceedings of the Eleventh Conference on Biocybernetics and Biomedical Engineering, 2, 1999, 842-846.

[2]. Lin R.H, an Intelligent Model for Liver Disease Diagnosis. Artificial Intelligence in Medicine, 47 (1), 2009, 53-62.

[3]. Rajeswari P and Reena G, Analysis of Liver Disorder using Data mining Algorithm. Global

Journal of Computer Science and Technology, 10 (14), 2010, 48-52

[4]. Ramana B.V, Babu M.S.P, and Venkateswarlu N.B, A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis. Global Journal of Database Management Systems, 3 (2), 2011, 101-114.

[5]. home.etf.rs/~vm/os/dmsw/Random%20Forest.pptx, last accessed 10/8/2015.

[6]. Jehad Ali et.al, "Random forest and decision trees ", IJCSI, Vol 9, No 3,pp272-278(2012).

[7]. Saaol times, Monthly magazine" Modifiable risk factors of heart disease", pp 6-10, July (2015).

[8]. Khan MG, "Heart disease diagnosis and therapy ", a practical approach, 2nd Edition Springer, pp544(2015).

[9]. Khan MG, "Heart disease diagnosis and therapy ", a practical approach,2nd Edition Springer, pp544(2015).

[10]. M.A. Jabbar, B L Deekshatulu, Priti chandra," classification of heart disease using artificial neural network and feature subset selection", GJCST, Vol13, issue 3,2013

[11]. https://en.wikipedia.org/wiki/Recurrent_neural_network

[12]. https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be

[13]. https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification

Cite this article as :

Gaurav Kumar Daharia, Mayank Tiwari, Priyanka Gupta, "Implementation of Healthcare Aid for Ailment Detection and Remedy using NLP and Full Stack", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 2, pp. 317-323, March-April 2021. Available at doi : https://doi.org/10.32628/CSEIT217261
Journal URL : https://ijsrcseit.com/CSEIT217261