 HEART DISEASE ANALYSIS

USING RNN

R

# 

# NAAN MUDHALVAN PROJECT

***Submitted by***

**R.SURUTHI (730321104056)**

**SIXTH SEMESTER**

**COMPUTER SCIENCE AND ENGINEERING**

BUILDERS ENGINEERING COLLEGE, KANGEYAM

**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY 2024**



**Name......................................................................................................................**

**Reg.No.................... ............... ............... ..............................................................**

**Branch....................................................................................................................**

**Subject code..........................................................................................................**

**Subject name.......................................................................................................**

**Year.......................................................Semester..................................................**

**BONAFIDE CERTIFICATE**

Certified that this is the bonafide record of work done by  **SURUTHI R (730321104056)** during the Academic year 2023- 24 for sixth Semester.

**STAFF -INCHARGE HEAD OF THE DEPARTMENT**

Submitted for the Naan Mudhalvan -**NM1009 Generative AI for Engineering** project held on………………..

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

I would like to express our deep and heartfelt thanks to our respected Chairman **Thiru. N. RAMALINGAM**, Honorable Secretary **Thiru C. K. VENKATACHALAM**, beloved Correspondent **Thiru S. ANANDAVADIVEL** and our responsible Treasurer **Thiru C. K.BALASUBRAMANIAM** for giving motivation and providing all necessary facilities for successful completion of the Internship project.

I take this opportunity to express our profound gratitude and deep regards to **Dr. C. VENKATESH, M.E.,Ph.D.,** Chief Executive Officer, Kangeyam Group of Institutions, for his constant encouragement throughout the course of our work.

I would like to express our gratitude to **Dr. S. RAMKUMAR, M.E., Ph.D.,** Principal, for the moral support and deeds in bringing out this Internship project successfully.

I express my sincere thanks to **Dr. S.D. PRABU RAGAVENDIRAN,** **M.E., Ph.D.,** Associate Professor & Head of the Department, Department of Computer Science and Engineering for encouraging a lot throughout this work.

I highly obliged to the Project Supervisor **Ms. P.SANTHIYA,** **M.E., (Ph.D).,** Assistant Professor, Department of Computer Science and Engineering for her guidance, suggestions and active encouragement for the fruitful completion of this project.

I would also like to thank and express our deep regards to all **Teaching faculty members and Non-teaching staff members** of our department. My sincere thanks to our **Parents** and **Friends** for their continuous encouragement and the untiring support rendered to us in all deeds and walks of this project.

# TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE NO.** |
| **1** | **ABSTRACT**  **INTRODUCTION** | **1** |
|  | 1.1 AIM |  |
|  | 1.2 PURPOSE |  |
| **2** | 1.3 USE CASES 1.4OBJECTIVES  **PROJECT DESCRIPTION** | **4** |
| **3** | 2.1 MODULE DESCRIPTION  **CONCLUSION AND FUTURE SCOPE**  3.1CONCLUSION  3.2 FUTURE SCOPE | **6** |
|  |  |  |
| **4** | **APPENDICES** |  |
|  | A1. SOURCECODE | 7 |
|  | A2. SCREENSHOTS | 10 |
| **5** | **REFERENCES** | **11** |
| **6** | **CERTIFICATES** | **15** |

**ABSTRACT**

This analysis presents an analysis of heart disease using a recurrent neural network (RNN). The RNN model is trained on longitudinal patient data to predict the onset and progression of heart disease. The analysis reveals significant correlations between various risk factors and the likelihood of developing heart disease, providing insights for early detection and preventive measures.

# CHAPTER 1 INTRODUCTION

* 1. **AIM**

The aim of heart disease analysis using a recurrent neural network (RNN) is to leverage the network's ability to process sequential data, such as time-series data from medical records or monitoring devices, to predict, diagnose, or classify heart diseases accurately. RNNs can learn patterns and relationships in the data over time, allowing for more effective analysis and prediction compared to traditional methods.

# USE CASES

* Sequential Data Processing
* Pattern Recognition
* Early Detection and Prediction
* Personalized Medicine
* Integration with Healthcare Systems
* Continuous Monitoring and Feedback

# PURPOSE

Analyzing heart disease using recurrent neural networks (RNNs) serves a multifaceted purpose in modern healthcare. One primary purpose is the early detection and diagnosis of heart conditions. By processing ECG data with RNNs, healthcare professionals can identify subtle deviations from normal heart rhythms indicative of various cardiovascular disorders, including arrhythmias, myocardial infarction, and heart failure.

Early detection enables timely intervention, potentially preventing adverse events and improving patient outcomes.By analyzing long-term ECG recordings or other physiological signals over time, RNN models can identify high-risk patterns associated with future cardiovascular events.

RNN-based analysis facilitates personalized medicine by capturing individual variations in heart function and response to treatment. By continuously monitoring and analyzing patient-specific data, such as heart rate variability, RNN models can adapt over time, providing dynamic insights into disease progression and treatment efficacy. This personalized approach enables healthcare providers to deliver targeted interventions and adjust treatment strategies in real-time, maximizing therapeutic benefits and minimizing adverse effects.

Overall, the purpose of heart disease analysis using recurrent neural networks extends beyond diagnosis and treatment—it encompasses early detection, risk stratification, personalized medicine, and medical research, ultimately aiming to improve patient outcomes, enhance healthcare delivery, and advance our understanding of cardiovascular health and disease.

# OBJECTIVES

The objectives of heart disease analysis using recurrent neural networks (RNNs) include:

1. Early Detection: Identify abnormalities in heart rhythms or other cardiac indicators at an early stage to facilitate timely intervention and prevent adverse events.

2. Diagnosis: Assist healthcare professionals in accurately diagnosing various cardiovascular disorders, such as arrhythmias, myocardial infarction, and heart failure, based on patterns detected in ECG data.

3. Risk Stratification: Assess the risk of future cardiovascular events by analyzing long-term ECG recordings or other physiological signals, enabling personalized risk assessment and preventive strategies.

4. Prognosis Assessment: Provide insights into disease progression and treatment response to optimize patient care and resource allocation, improving patient outcomes and quality of life.

5. Personalized Medicine: Tailor treatment plans and preventive strategies based on individual patient risk profiles and physiological characteristics, maximizing therapeutic benefits and minimizing adverse effects.

6. Medical Research: Contribute to the advancement of cardiovascular medicine by uncovering novel biomarkers, physiological mechanisms, and therapeutic targets through the analysis of large-scale datasets and advanced machine learning techniques.

7. Innovation: Drive the development of innovative diagnostic tools, predictive models, and therapeutic interventions to improve the prevention, diagnosis, and management of heart disease on a global scale.

# CHAPTER 2

# PROJECT DESCRIPTION

This project aims to develop a predictive model using recurrent neural networks (RNNs) to analyze heart disease. Heart disease is a prevalent health concern globally, and early detection plays a vital role in effective management and prevention.

The project will involve:

1. Data Collection: Gather a comprehensive dataset containing various attributes related to heart health, including demographic information, medical history, lifestyle factors, and diagnostic test results.

2. Data Preprocessing: Cleanse and preprocess the dataset to handle missing values, normalize features, and encode categorical variables.

3. Model Development: Implement an RNN architecture suitable for sequential data analysis. Experiment with different RNN variants such as LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) to capture temporal dependencies effectively.

4. Training and Validation: Split the dataset into training and validation sets. Train the RNN model on the training data while monitoring performance on the validation set to prevent overfitting.

5. Performance Evaluation: Evaluate the trained model using appropriate metrics such as accuracy, precision, recall, and F1-score. Conduct thorough analyses to understand the model's strengths and limitations.

6. Deployment and Integration: Deploy the trained RNN model into a user-friendly interface, allowing healthcare professionals to input patient data and obtain predictions regarding heart disease risk.

# 2.1 MODULE DESCRIPTION

This module focuses on the development of a recurrent neural network (RNN) architecture tailored for heart disease analysis. The RNN will be designed to effectively capture temporal dependencies within sequential patient data, enabling accurate prediction of heart disease risk. The module encompasses various stages from data preprocessing to model evaluation, with the goal of providing accurate predictions for proactive healthcare management.

**1. Data Preprocessing Module:**

Cleanses and preprocesses raw datasets containing patient demographics, medical history, diagnostic tests, and other relevant factors.

Handles missing values, normalizes features, and encodes categorical variables to prepare data for model training.

**2. RNN Model Architecture Selection:**

Explores different RNN variants such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) to capture temporal dependencies effectively.

Considers model complexity, memory efficiency, and computational resources for selecting the most suitable architecture.

**3. Training and Validation Module:**

Implements the chosen RNN architecture using deep learning frameworks like TensorFlow or PyTorch.

Divides the dataset into training, validation, and possibly testing sets for model training and evaluation.

Monitors training progress, adjusts hyperparameters, and prevents overfitting through techniques like regularization and dropout.

**4. Model Evaluation and Performance Metrics:**

Evaluates the trained RNN model using appropriate metrics such as accuracy, precision, recall, and F1-score.

Conducts cross-validation and statistical analysis to assess the model's generalization capabilities and robustness.

**5. Deployment and Integration:**

Deploys the trained RNN model into a user-friendly interface or integrates it into existing healthcare systems for practical use.

Ensures compatibility with data privacy regulations and interoperability with electronic health records (EHR) systems.

**6. Documentation and Reporting:**

Documents the entire process, including data preprocessing steps, model architecture details, training procedures, and evaluation results.

Prepares a comprehensive report summarizing key findings, insights, and recommendations for stakeholders and future research directions.

# CHAPTER 3

# CONCLUSION AND FUTURE SCOPE

**3.1 CONCLUSION**

The application of recurrent neural networks (RNNs) for heart disease analysis offers promising avenues for proactive healthcare management. By leveraging the temporal dependencies present in sequential patient data, RNN models can accurately predict heart disease risk, contributing to early detection and intervention strategies. Through meticulous data preprocessing, model selection, training, and evaluation, RNN-based approaches empower healthcare professionals with powerful tools for improving patient outcomes and optimizing resource allocation. However, continual refinement and validation of these models are essential to ensure reliability and generalizability across diverse patient populations. Overall, RNN-based heart disease analysis represents a significant step towards personalized and data-driven healthcare solutions, with the potential to revolutionize cardiovascular care in the future.

# 3.2 FUTURE SCOPE

The future scope of heart disease analysis using recurrent neural networks (RNNs) holds immense potential for advancing healthcare outcomes and research in cardiovascular medicine.

Here are some avenues for future exploration:

* Enhanced Model Architectures
* Multi-modal Data Integration
* Longitudinal Analysis
* Personalized Risk Assessment
* Explainable AI
* Real-time Monitoring and Intervention
* Population-level Studies
* Integration with Telemedicine
* Clinical Trials and Drug Development
* Ethical and Regulatory Considerations

**A1.SOURCE CODE**

**Importing Libraries:**

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Activation

import numpy as np

import pandas as pd

import seaborn as sns

import numpy as np

from matplotlib import pyplot as plt

"""   
#Dataset link:

https://cdn.scribbr.com/wp-content/uploads//2020/02/heart.data\_.zip?\_ga=2.217642335.893016210.1598387608-409916526.1598387608

#Heart disease

The effect that the independent variables biking and smoking

have on the dependent variable heart disease

"""

**Reading dataset:**

df = pd.read\_csv('data/heart\_data.csv')

print(df.head())

**Ploting the Model:**

df = df.drop("Unnamed: 0", axis=1)

#A few plots in Seaborn to understand the data

sns.lmplot(x='biking', y='heart.disease', data=df)

sns.lmplot(x='smoking', y='heart.disease', data=df)

x\_df = df.drop('heart.disease', axis=1)

y\_df = df['heart.disease']

x = x\_df.to\_numpy()

y = y\_df.to\_numpy()

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=42)

**Compile the Model:**

# Build the network

# sgd = optimizers.SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)

model = Sequential()

model.add(Dense(2, input\_dim=2, activation='relu'))

model.add(Dense(1))

model.compile(loss='mean\_squared\_error', optimizer='adam')

print(model.summary())

history = model.fit(X\_train, y\_train ,verbose=1, epochs=1000,

validation\_data=(X\_test, y\_test))

# Predict

prediction\_test = model.predict(X\_test)

print(y\_test, prediction\_test)

print("Mean sq. errror between y\_test and predicted =", np.mean(prediction\_test-y\_test)\*\*2)

#plot the training and validation accuracy and loss at each epoch

loss = history.history['loss']

val\_loss = history.history['val\_loss']

epochs = range(1, len(loss) + 1)

plt.plot(epochs, loss, 'y', label='Training loss')

plt.plot(epochs, val\_loss, 'r', label='Validation loss')

plt.title('Training and validation loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()

##############################

# Print weights

for layer\_depth, layer in enumerate(model.layers):

weights = layer.get\_weights()[0]

biases = layer.get\_weights()[1]

print('\_\_\_\_\_\_\_\_\_\_\_ Layer', layer\_depth, '\_\_\_\_\_\_\_\_\_\_')

for toNeuronNum, bias in enumerate(biases):

print(f'Bias to Layer{layer\_depth+1}Neuron{toNeuronNum}: {bias}')

for fromNeuronNum, wgt in enumerate(weights):

for toNeuronNum, wgt2 in enumerate(wgt):

print(f'Layer{layer\_depth}, Neuron{fromNeuronNum} to Layer{layer\_depth+1}, Neuron{toNeuronNum} = {wgt2}')

#######################################################################

"""

As the weights change for each run let us use the following weights for our calculations

\_\_\_\_\_\_\_\_\_\_\_ Layer 0 \_\_\_\_\_\_\_\_\_\_

Bias to Layer1Neuron0: 4.4128947257995605

Bias to Layer1Neuron1: 4.5146260261535645

Layer0, Neuron0 to Layer1, Neuron0 = -0.08574138581752777

Layer0, Neuron0 to Layer1, Neuron1 = -0.059531815350055695

Layer0, Neuron1 to Layer1, Neuron0 = 0.1630137860774994

Layer0, Neuron1 to Layer1, Neuron1 = -0.015843335539102554

\_\_\_\_\_\_\_\_\_\_\_ Layer 1 \_\_\_\_\_\_\_\_\_\_

Bias to Layer2Neuron0: 2.504946708679199

Layer1, Neuron0 to Layer2, Neuron0 = 1.4296010732650757

Layer1, Neuron1 to Layer2, Neuron0 = 1.1467727422714233

"""

x0 = 65.1292

x1 = 2.21956

hidden0 = max(0, ((x0\*-0.08574138)+(x1\*0.163013786)+(4.4128947)))

hidden1 = max(0, ((x0\*-0.0595318)+(x1\*-0.0158433)+(4.514626)))

output = max(0, ((hidden0\*1.4296)+(hidden1\*1.14677)+(2.504947)))

print(output)

**A2.SCREENSHOT**



