International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Recognition RNN Based Heartbeat Sound Analysis with Django Integration

Dr.G.DhanaLakshmi [1a], Krishnapriya M[2a], Monisha N V[2b]

[1] AssociateProfessor,Department of Information Technology, Panimalar Engineering College
[2] IV Students, Department of Information Technology, Panimalar Engineering College
gdhanalakshmipecit@gmail.com| krrishnapriya170@gmail.com | monishanv2003@gmail.com

How to Cite this Article:

M, K. & V, M. N. (2026). Recognition RNN Based Heartbeat Sound Analysis with Django Integration. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(05). https://doi.org/10.55041/ijcope.v2i5.421

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.

OPEN ACCESS

https://doi.org/10.55041/ijcope.v2i5.421

***Abstract*--**This project presents an innovative approach to heartbeat audio classification using Recurrent Neural Networks (RNNs) integrated with the Django framework. The primary aim is to develop an efficient and accurate system for classifying heartbeat sounds to aid in the early detection and diagnosis of cardiac conditions. The system leverages RNNs, which are particularly suited for processing sequential data, to analyse and classify heartbeat audio recordings. The Django framework facilitates seamless integration, providing a robust and scalable web application for data management, model deployment, prediction. The RNN model is trained on a diverse dataset of heartbeat audio recordings, enabling it to recognize various cardiac anomalies. The proposed system demonstrates high accuracy and reliability, making it a valuable tool for healthcare professionals. Additionally, the integration with Django ensures that the system can be easily accessed and utilized in clinical settings, promoting widespread adoption and improving patient outcomes.

Keywords--Recurrent Neural Networks (RNNs), heartbeat audio classification, Django integration, cardiac conditions, sequential data processing, model deployment, patient outcomes.

I. INTRODUCTION

In recent years, the analysis of heartbeat sounds has gained considerable attention in the field of medical diagnostics due to its immense potential in the early detection of cardiovascular abnormalities. Cardiovascular diseases (CVDs) remain a leading cause of morbidity and mortality worldwide, and timely diagnosis is crucial for effective intervention and treatment. Traditional analysis techniques, although widely used, rely heavily on the subjective interpretation of healthcare professionals, which can lead to variability in diagnosis. With advancements in artificial intelligence (AI) and machine learning (ML), particularly in deep learning architectures such as Recurrent Neural Networks (RNNs), automated analysis of heartbeat sounds has emerged as a promising solution for improving diagnostic accuracy and efficiency. This project focuses on developing an RNN-based system for heartbeat sound analysis, enabling automated classification and anomaly detection. By utilizing digital stethoscope recordings, the system processes heartbeat audio signals using advanced deep learning techniques, particularly Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU), both of which are well-suited for capturing the temporal dependencies inherent in sequential medical data. The analysis begins with pre-processing the raw heartbeat recordings to remove noise and enhance signal clarity, ensuring that the extracted features represent meaningful physiological characteristics. Librosa, a powerful Python library for audio analysis, is employed for feature extraction, transforming the raw audio data into a structured format suitable for deep learning models.

Once the features are extracted, an RNN-based model is trained to distinguish between normal and abnormal heart sounds, allowing for the early detection of conditions such as arrhythmias, murmurs, and other cardiac irregularities. The model's performance is evaluated using key metrics such as accuracy, sensitivity, and specificity to ensure its reliability and effectiveness in real-world applications. Additionally, the integration of Django for web application development enables seamless deployment of the system, providing a user- friendly interface for healthcare professionals and researchers to upload and analyze heartbeat recordings with ease.

By combining state-of-the-art deep learning methodologies with robust audio signal processing techniques, this project aims to bridge the gap between AI and clinical diagnostics. The proposed system not only enhances diagnostic accuracy but also democratizes access to advanced cardiovascular analysis tools, particularly in resource-limited settings where expert cardiologists may not always be available. The convergence of audio analysis, deep learning, and web-based deployment positions this project at the forefront of digital healthcare innovation, offering a scalable and

accessible solution for cardiovascular disease screening. This research contributes to the ongoing evolution of AI- driven medical diagnostics, paving the way for more efficient, accurate, and early detection of cardiac disorders, ultimately leading to improved patient outcomes and reduced healthcare burdens.

II. LITERATURE SURVEY

Title: EfficientNet Optimization on jiffs Sound Bracket
Author: Husni Fadhilah Dhiya Ul Haq
Year: 2021
Abstract-- presently, utmost deaths are caused by heart complaint. twinkle sound analysis is a straightforward way to diagnose heart complaint and eventuality for early discovery of heart conditions. still, echocardiographic bracket from twinkle sound has always been a challenge in point birth. In this paper, EfficientNet is optimized for twinkle sound bracket. We optimized four EfficientNet infrastructures and thick layers number and powerhouse settings. twinkle sounds were converted in spectrogram format as input. Data set B was applied in this study comported of three orders normal, murmur, and extrasystole heart sound. The stylish armature was EfficientNet B0 with four thick layers and powerhouse in the test dataset with 82 delicacy.

Title: Heart Rate Discovery and Bracket from Speech Spectral Features Using Machine Learning
Author: Mohammed ZUBAIR
Year: 2021
Abstract-- dimension of vital signs of the mortal body similar as heart rate, blood pressure, body temperature and respiratory rate is an important part of diagnosing medical conditions and these are generally measured using medical outfit. In this paper, we propose to estimate an important vital sign – heart rate from speech signals using machine

literacy algorithms. Being literature, observation and experience suggest the actuality of a correlation between speech characteristics and physiological, cerebral as well as emotional conditions. In this work, we estimate the heart rate of individualities by applying machine literacy grounded retrogression algorithms to Mel frequence cepstrum portions, which represent speech features in the spectral sphere as well as the temporal variation of spectral features. The estimated heart rate is compared with factual dimension made using a conventional medical device at the time of recording speech. We gain estimation delicacy close to 94 between the estimated and factual measured heart rate values. double bracket of heart rate as 'normal' or 'abnormal' is also achieved with 100 delicacy. A comparison of machine literacy algorithms in terms of heart rate estimation and bracket delicacy is also presented. Heart rate dimension using speech has operations in remote monitoring of cases, professional athletes and can grease telemedicine.

Title: twinkle Sound point birth and Bracket

Author: Fariha Chowdhury Bibrity

Year: 2019

Abstract-- Heart conditions has ranked top as the cause of death encyclopedically. The harsh verity is, in this time it's hard to get proper medical treatment in proper time and still it's expensive. Now the only light of stopgap is coming from technology. Heart sound is one of the oldest ways to judge the condition of the heart. This paper shows the issues from a set of uprooted features of twinkle sound by applying the classifier Naıve Bayes, Neural Network, Decision Tree, SVM, Logistic Regression and Nearest Neighbour. Experimental results show that SVM carried the loftiest delicacy(i.e., 76) for normal and abnormal twinkle bracket, ANN(i.e., 83) for normal and murmur bracket and Nearest Neighbour(i.e., 73) for normal and extrasystole bracket compared to other machine learning algorithms. This exploration includes comparing the results from all the algorithms and chancing the stylish possible set of data and algorithms. This machine learning fashion contributes to the development of heart complaint related inquiries and developing more effective machines to descry heart conditions directly in short time.

Title: A Comprehensive Survey on Heart Sound Analysis in the Deep literacy period.

Author: Zhao Ren, University of Bremen.

Time: 2023

Abstract-- Heart sound auscultation has been applied in clinical operation for early webbing of cardiovascular conditions. Due to the high demand for auscultation moxie, automatic auscultation can help with supplementary opinion and reduce the burden of training professional clinicians. nonetheless, there's a limit to classic machine literacy's performance enhancement in the period of big data. Deep literacy has outperformed classic machine literacy in numerous exploration fields, as it employs more complex model infrastructures with a stronger capability of rooting effective representations. also, it has been successfully applied to heart sound analysis in the once times. As utmost review works about heart sound analysis were carried out before 2017, the present check is the first to work on a comprehensive overview to summarise papers on heart sound analysis with deep literacy published in 2017 – 2022. This work introduces both classic machine literacy and deep literacy for comparison, and farther offer perceptivity about the advances and unborn exploration directions in deep literacy for heart sound analysis.

Title: operation of Machine Learning ways for Heart Sound Recording Bracket Author Anatoly Yakovlev, yakovlev

Year: 2023

Abstract-- Ultrasound and electro cardiogram(ECG) are standard, estimable styles for heart complaint opinion, but due to precious and limited availability, typical primary opinion are performed by largely trained croakers with stethoscope sounds. Due to global deficit of healthcare providers, there's growing interest for a cheaper, more ubiquitous volition. The envisaged thing is to determine, from a single short 5- 120 sec precordial heart sound recording, whether the case should be appertained to expert opinion. similar heart sound recordings can come from colorful clinical or nonclinical surroundings, including at-home recordings with a smartphone to in- sanitarium recordings by a nanny . For opinion to be successfully automated, heart sound waveforms must be segmented into applicable S1, systole, S2, and diastole phases of the heart cycle. In this paper, logistic retrogression hidden semi-Markov model is used for waveform segmentation. Features are also uprooted from the segmented waveforms and used in confluence with croaker - handed bracket markers, to train supervised machine literacy models to identify heart abnormality in patient test data. colorful supervised machine literacy models are explored, enforced, and compared for performance.

Title: Reopened intermittent Unit- grounded heart sound analysis for heart failure webbing.
Author: Gao S, Zheng Y, Guo X.
Year: 2020
Abstract-- The proposed a new HF webbing frame grounded on reopened intermittent unit(GRU) model in this study. The logistic retrogression- grounded retired semi-Markov model was espoused to member HS frames. regularized frames were taken as the input of the proposed model which can automatically learn the deep features and complete the HF webbing without de-nosing and hand- drafted point birth.

Title: Reopened intermittent Unit- grounded heart sound analysis for heart failure webbing.
Author: Gao S, Zheng Y, Guo X.
Year: 2020

Abstract — The proposed a new HF webbing frame grounded on reopened intermittent unit(GRU) model in this study. The logistic retrogression- grounded retired semi-Markov model was espoused to member HS frames. regularized frames were taken as the input of the proposed model which can automatically learn the deep features and complete the HF webbing without de-nosing and hand- drafted point birth.

Title: Deep literacy- Grounded Heart Sound Analysis for Left Ventricular Diastolic Dysfunction opinion.
Author: Yang Y, Guo XM, Wang H, Zheng YN.
Time: 2021
Abstract — The aggravation of left ventricular diastolic dysfunction(LVDD) could lead to ventricular redoing, wall stiffness, reduced compliance, and progression to heart failure with a saved ejection bit. Anon-invasive system grounded on convolutional neural networks(CNN) and heart sounds(HS) is presented for the early opinion of LVDD in this paper. A deep convolutional generative inimical networks(DCGAN) model-grounded data addition(DA) system was proposed to expand a HS database of LVDD for model training. originally, the pre-processing of HS signals was performed using the bettered sea denoising system. Secondly, the logistic retrogression grounded retired semi-Markov model was employed to member HS signals, which were latterly converted into spectrograms for DA using the short- time Fourier transfigure(STFT). Eventually, the proposed system was compared with VGG- 16, VGG- 19, ResNet- 18, ResNet- 50, Dense Net- 121, and AlexNet in terms of performance for LVDD opinion. The result shows that the proposed system has a reasonable performance with an delicacy of 0.987, a perceptivity of 0.986, and a particularity of 0.988, which proves the effectiveness of HS analysis for the early opinion of LVDD and demonstrates that the DCGAN- grounded DA system could effectively compound HS data.

III. EXISTING SYSTEM

Heartbeat sound analysis has been an essential tool in diagnosing cardiovascular diseases. Traditionally, heart sounds are evaluated by physicians using a stethoscope, a method that requires considerable expertise and experience. While manual auscultation has been the gold standard for many years, it is subject to human error and can be inconsistent, especially in identifying subtle or complex abnormalities in heartbeats. Over the past few decades, advancements in signal processing and machine learning have led to the development of several automated systems aimed at improving the accuracy and reliability of heart sound analysis.

1. Traditional Signal Processing Methods

Before the advent of deep learning, heart sound analysis primarily relied on traditional signal processing techniques. These methods often involved extracting features such as frequency, amplitude, and wave patterns from the acoustic signals of heartbeats. Common techniques included:

-Fourier Transform and Spectral Analysis: These methods focus on analyzing the frequency domain of heart sound signals to identify specific patterns, such as murmurs or abnormal rhythms. However, they often struggle to detect complex, time-varying features of heart sounds and may fail to capture long-term dependencies.

-Wavelet Transform: Wavelet-based techniques have been used to decompose the heart sound signals into multiple frequency bands, enabling the detection of high-frequency murmurs or low-frequency sounds associated with heart conditions. Despite their versatility, these methods still rely heavily on manual feature extraction, which can be error-prone and time-consuming.

While these traditional methods provided some level of success in heart sound analysis, they generally lacked the ability to model the intricate, sequential nature of heart sounds effectively, especially in the context of complex cardiovascular diseases.

2. Machine Learning Approaches

With the rise of machine learning, several systems have emerged that use algorithms like Support Vector Machines (SVM), Random Forests, and K-Nearest Neighbours (KNN) to classify heart sounds. These systems typically require pre-processing to extract relevant features, such as pitch, duration, and frequency patterns, which are then fed into the classifier. Some advantages and limitations of these methods include:

-Advantages: Machine learning models can automate the classification process, reducing the reliance on human expertise and making the system more accessible.

-Limitations: Despite improvements, these systems still face challenges with feature selection and classification accuracy, especially when it comes to classifying rare or subtle heart abnormalities. Additionally, they are often limited by the features chosen and may not fully exploit the raw temporal information inherent in heart sounds.

3. Limitations of Existing Systems

While the existing systems and methods have made significant strides in automating heart sound analysis, there are several key limitations**:**

-Dependence on Pre-processing: Many methods, including both traditional signal processing and machine learning approaches, still require complex pre-processing steps, such as manual feature extraction or transformation into the frequency domain. This can introduce noise or loss of critical information.

- Limited Generalization: Most existing systems are trained on specific datasets, which may not generalize well to new, unseen data or to heart sounds recorded from diverse populations with varying conditions.

- Real-Time Analysis: While some methods provide accurate classification, real-time heart sound analysis remains a challenge due to computational constraints and the need for fast processing in clinical environments.

4. Our Contribution: RNN-based Heartbeat Sound Analysis

In contrast to these existing systems, our proposed approach leverages Recurrent Neural Networks (RNNs), specifically LSTM and GRU, to process raw heart sound signals directly. By utilizing the temporal dependencies present in heart sounds, our system can classify heartbeats accurately without the need for manual feature extraction. Furthermore, our approach offers the following advantages:

-End-to-End Learning: The RNN-based model can automatically learn from raw acoustic signals, eliminating the need for manual feature extraction and reducing the risk of information loss.

-Improved Temporal Modeling: By using RNN architectures, our system can better capture long-term dependencies in heart sound signals, which is essential for detecting complex conditions such as murmurs, arrhythmias, and other irregular heart rhythms.

-Scalability: Our model can be trained on large, diverse datasets and has the potential for real-time analysis, making it suitable for both clinical and remote health monitoring applications.

IV. PROPOSED SYSTEM

The proposed system for Recognition RNN-Based Heartbeat Audio Classification with Django Integration leverages Recurrent Neural Network (RNN) architectures to analyze and classify heartbeat audio signals for medical diagnostics. This system collects audio recordings of heartbeats, processes them to extract relevant features, and uses a trained RNN model to classify the heartbeats into categories. The Django framework serves as the backbone for the web application, facilitating user interaction, data management, and visualization of the classification results. Users can upload audio files through the Django-based interface, receive immediate feedback on the classification, and access a comprehensive database of recorded results and associated. This integration aims to provide an efficient and accessible tool for early detection of Heartbeat conditions, improving diagnostic accuracy and patient outcomes.

1. SYSTEM ARCHITECTURE:

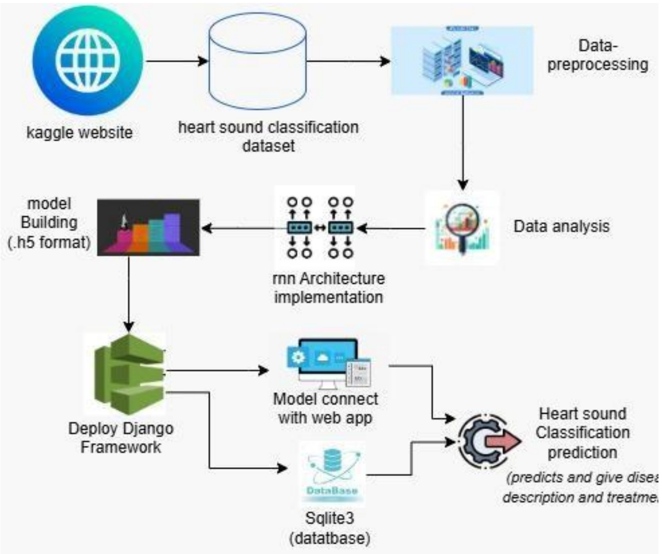


Figure 1. System Architecture

1.1.System Overview :

The architecture of the system consists of two main components:

- **Recognition RNN-based Heartbeat Sound Analysis:** This component processes and analyzes heartbeat sounds to detect anomalies or classify them into specific categories.
- **Django Integration:** This serves as the web application interface that communicates with the analysis component, displays results to the user, and manages backend processes.

1.1.1.Component Breakdown :

Heartbeat Sound Data Acquisition:

- **Input:** The system starts by capturing heartbeat sounds from users, typically through a microphone or stethoscope connected to the user's device.
- **Pre-processing**: The raw sound data is pre-processed to remove noise, normalize volume levels, and segment it into windows for analysis. This may include techniques like filtering, denoising, or converting the sound into a spectrogram representation.

1.1.2.RNN-Based Heartbeat Sound Recognition: Recurrent Neural Network (RNN) is the core model that is responsible for analyzing the heartbeat sound. The RNN, likely using LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) units, is trained on labelled heartbeat sound data to classify different heartbeats, detect irregularities (like murmurs or arrhythmias), or evaluate heart conditions based on sound patterns.

- **Model Training:** The RNN model is trained using a labelled dataset of heartbeat sounds. Various audio features such as Mel-frequency cepstral coefficients (MFCCs), spectral features, or raw waveforms are extracted from the audio, which is then fed into the RNN.
- **Output:** The RNN outputs a classification or recognition result (e.g., "Normal Heartbeat," "Arrhythmia," "Murmur," etc.), based on the learned patterns in the sound data.

1.1.3.Model Integration with Django Backend:

- **Django Framework:** The Django framework is used to build the backend of the application, where the analysis and processing of heartbeat sounds occur.
- **Web Interface:** The user interface (UI) is built using Django templates, allowing users to upload heartbeat sound files or interact with the system in real-time (if connected to a live microphone).
- **Backend Logic:** When a user uploads or records a heartbeat sound, the Django backend processes this file and sends it to the RNN model for analysis. The backend handles model inference, calling the trained RNN model to classify the sound and return the result.
- **Results Presentation:** Once the RNN processes the sound and returns a classification result, Django presents it to the user on a web page. This could include detailed information like the type of heartbeat, confidence levels, or suggested follow-up actions (e.g., visit a doctor).

1.1.4.Django-Integrated API for Communication:

- **API Endpoints:** The Django application exposes API endpoints (possibly using Django REST Framework) for communication between the front-end and the backend. These endpoints handle tasks such as uploading audio, receiving analysis results, and managing user data.
- **Integration with Front-End:** The frontend could be built using HTML, CSS, and JavaScript or a front-end framework (like React or Vue.js) that interacts with the Django backend via API calls to submit heartbeat sounds and display results.
- **Real-Time Analysis (Optional):** For real-time recognition, Django can implement WebSockets or other real-time communication technologies to deliver results instantly when a heartbeat sound is detected.

1.1.5.Data Storage:

- **Database:** Django's Sqlite (database) system is used to manage data storage. This could store:

a) User information (e.g., history of analysis, personal data)

b) Audio files (either raw sound or pre-processed features)

c) Model analysis results and logs

- **File Storage:** Heartbeat audio files can be stored in the server's local storage or cloud-based storage like AWS S3 for easy access and retrieval.

1.1.6.Model Training and Deployment:

- **Training Environment:** The RNN model is trained offline on a powerful server or cloud platform that supports machine learning workloads. It is trained using libraries such as TensorFlow, Keras, or PyTorch.

1.1.7.Model Deployment: Once trained, the RNN model is exported and integrated into the Django backend, possibly using frameworks like TensorFlow Serving, Flask, or FastAPI for serving the model. This allows Django to make requests to the model and retrieve analysis results.Security & Privacy:

- Authentication: Django's authentication system ensures secure user login, profile management, and access control. Only authorized users can upload heartbeat sounds or view sensitive results.
- Data Protection: Given the health-related nature of the data, security measures such as encryption, secure file uploads, and compliance with data protection regulations (like HIPAA or GDPR) should be implemented.

1.2.REQUIREMENTS:Hardware Requirements:

- Processor : Pentium IV/III
- Hard disk: minimum 80 GB
- RAM : minimum 8 GB

1.3.Software Requirements:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk- learn, pandas, numpy, matplotlib and seaborn.

- Operating System: Windows
- Tool: Anaconda with Jupyter Notebook

V. IMPLEMENTATION:

1. Data Analysis

In the realm of heartbeat sound analysis, utilizing Recurrent Neural Networks (RNNs), particularly with architectures like SimpleRNN and Long Short-Term Memory (LSTM) networks, offers robust solutions for pattern recognition in audio data. Libraries such as Librosa facilitate effective audio processing, allowing the extraction of relevant features from heartbeat sounds, such as tempo, pitch, and rhythm. By employing these features, RNNs can learn temporal dependencies, crucial for accurately distinguishing between various heartbeats, including normal and abnormal patterns.

- **Dataset Exploration**:

o Overview of the dataset(s) used (e.g., heart sound recordings, types of heart sounds: normal, murmur, etc.).

o Data distribution: How many samples per class, how balanced or imbalanced the dataset is.

- **Statistical Analysis**:

o Basic statistics of the dataset (e.g., average duration of heartbeats, common frequencies in heart sound).

o Visualizations like histograms, spectrograms, or waveform plots of heart sounds to better understand the data.

- **Feature Engineering**:

o **Feature Extraction**: Detailed description of features like MFCC (Mel Frequency Cepstral Coefficients), spectral features, or raw waveform.

o **Time-Domain and Frequency-Domain Features**: Insights into the extraction of these features and their relevance to heart sound analysis.

o **Feature Normalization/Standardization**: Methods to scale features for model training, like min-max scaling or z-score normalization.

By utilizing the strengths of RNNs, particularly LSTM's capability to remember long-term dependencies, this approach can improve diagnostic accuracy and early detection of potential heart-related issues. Overall, this combination of advanced machine learning techniques and web technology represents a significant step forward in digital health applications.

The below 2 different algorithms are compared:

- Simple RNN Architecture
- Long-Short term memory networks

2. RNN/LSTM Implementation

A **Simple Recurrent Neural Network (Simple RNN)** is a type of artificial neural network designed to recognize patterns in sequences of data, such as time-series data, text, speech, or video. Unlike traditional feedforward neural networks, RNNs have a temporal dynamic behaviour, meaning they are capable of remembering past inputs due to the presence of loops in their architecture. This makes RNNs ideal for tasks that involve sequential data where order and temporal dependencies are important.

- **Recurrent Neural Network (RNN) Basics**:

○ Overview of RNNs: How RNNs work with sequential data (heart sound signals).

○ **Challenges with Basic RNNs**: Issues like vanishing gradients and long-range dependency learning.

- **LSTM (Long Short-Term Memory)**:

○ Introduction to LSTM networks: Why LSTM is chosen over basic RNNs (solves vanishing gradient problem, better for long sequences).

○ **LSTM Architecture**: Description of gates (forget, input, output) and how they help LSTM retain long-term dependencies.

○ **Model Architecture**:

Input Layer: Shape and preprocessing of the input data (e.g., one-dimensional heart sound signal, spectrogram, or feature vector).

○ **Hidden Layers**: Number of LSTM layers, units in each layer, and activation functions (usually tanh and sigmoid).

○ **Output Layer**: Softmax or sigmoid activation (for multi-class or binary classification).

○ **Dropout and Regularization**: Methods to prevent overfitting in LSTM (e.g., Dropout, L2 regularization).

- **Training LSTM Model**:

○ **Loss Function**: Cross-entropy for classification tasks (binary or multi-class).

○ **Optimizer**: Adam optimizer or RMSProp to adapt the learning rate.

○ **Hyperparameter Tuning**: Fine-tuning the learning rate, batch size, number of LSTM units, etc.

- **Performance Evaluation**:

○ Accuracy, Precision, Recall, F1-score for evaluating model performance.

○ Confusion matrix for multi-class or binary classification tasks.

○ Graphs showing training and validation loss, accuracy over epochs.

3. Deployment

Django is a micro web frame written in Python. It is classified as a micro-framework because it does not need particular tools or libraries. It has no database abstraction caste, form evidences, or any other factors where pre-existing third-party libraries gives common functions.

- **Model Export and Conversion**:

○ **Exporting Model**: Saving the trained model using libraries like TensorFlow's model. save() or PyTorch'storch.save() to produce a serial interpretation of the model.

○ **Model Conversion**: If deploying on mobile or embedded devices, converting the model to formats like TensorFlow Lite.

- **Integration with Web/Mobile Operations**:

○ **Backend Garcon**: Using frameworks like Flask or FastAPI to deploy the model on a server for real-time heart sound analysis.

○ **API Endpoint**: Exposing the model via REST API so it can be penetrated from a web or mobile operation function.

○ **Real-time Prognostications**: Integrating the API for real-time heart sound analysis (e.g., uploading a recording of heartbeats and receiving a diagnosis or classification result for the given model).

- **Optimization for Deployment**:

○ **Model Compression**: Methods like quantization, pruning, or knowledge distillation to reduce the model size and ameliorate conclusion speed for deployment in resource-constrained environments (mobile/embedded devices).

○ **Edge Deployment**: Deploying the model to edge bias (e.g., mobile phones or IoT bias for continuous heart sound monitoring).

○ **Latency and Throughput Optimization**: Ensuring the model performs well with minimal quiescence, especially for real-time operations.

- **Monitoring and Conservation**:

○ **Model Drift**: Monitoring the performance of the stationed model over time and detecting if the model performance degrades (e.g., due to changes in the input data distribution).

o **Model Retraining**: Setting up a process for periodic retraining of the model using new data or user feedback to maintain its delicacy.

- **Stoner Interface**:

o Design considerations for an intuitive interface for stoners to upload heart sound data (e.g., through a smartphone or web app) and admit bracket results.

o Visualizations (e.g., displaying heart sound waveforms, spectrograms) along with the model's affair (e.g., normal or abnormal twinkle classification).

VI. ALGORITHM:

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are commonly used for sequence-based tasks like speech, language, and audio analysis because they can retain information from previous time steps in the sequence. Below is an overview of both algorithms and their implementation steps.

1. Recurrent Neural Network (RNN) Algorithm:

RNNs are a type of neural network designed for sequence prediction tasks. They have a loop within their hidden layers, which allows them to maintain a memory of previous inputs in the sequence.

RNN Architecture:

- **Input Sequence:** A sequence of input data, x1,x2,...,xtx_1, x_2, ..., x_t, where each input xtx_t is associated with a time step.
- **Hidden State:** The hidden state hth_t at time tt is calculated as:

ht=Activation(Wh·ht−1+Wx·xt+b)h_t = \text{Activation}(W_h \cdot h_{t-1} + W_x \cdot x_t + b)

Where:

a) WhW_h is the weight matrix for the hidden state.

b) WxW_x is the weight matrix for the input.

c) bb is the bias term.

The output at each time step can be calculated as: yt=Wy·ht+byy_t = W_y \cdot h_t + b_y

- **Training:** The network is trained by minimizing the loss function, such as Mean Squared Error (MSE) for regression tasks or Cross-Entropy for classification tasks. **RNN Training Process**:

i) Initialize weights: Randomly initialize the weights and biases.

ii) Forward Pass: Process the sequence step-by-step and compute the hidden states and outputs.

iii) Loss Calculation: Calculate the loss between predicted outputs and actual labels.

iv) Backpropagation: Compute gradients using backpropagation through time (BPTT) and update weights.

v) Repeat: Repeat steps 2 to 4 for a number of epochs until the model converges.

2. Long Short-Term Memory (LSTM) Algorithm:

LSTM is a special type of RNN designed to address the vanishing gradient problem of standard RNNs, which makes it difficult for them to learn long-range dependencies in sequences. LSTMs introduce memory cells that store information for longer periods, making them more suitable for tasks involving long-term dependencies.

Key Concepts of LSTM:

- **Cell State:** The cell state acts as the memory of the LSTM. It carries information across time steps and is modified by gates that control the flow of information.
- **Gates:**

a) **Forget Gate:** Decides what information to throw away from the cell state. It outputs a value between 0 and 1 for each number in the cell state.

b) **Input Gate:** Decides what new information to store in the cell state.

c) **Output Gate:** Decides what information from the cell state to output.

LSTM Architecture:

1. Forget Gate:

ft=σ(Wf·[ht−1,xt]+bf)f_

t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)

The forget gate takes the previous hidden state ht−1h_{t- 1} and the current input xtx_t, and applies a sigmoid activation function to determine which parts of the cell state to forget.

2. Input Gate:

it=σ(Wi·[ht−1,xt]+bi)

i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)

The input gate controls how much new information is added to the cell state. It uses a sigmoid activation to decide which information to update.

3. CandidateCellState:

C~t=tanh(WC·[ht−1,xt]+bC)\tilde{C}

t=\tanh(W_C \cdot [h{t-1}, x_t] + b_C)

This computes the candidate cell state, which represents the new information to be stored in the cell state.

4. Update Cell State:

Ct=ft·Ct−1+it·C~tC_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t

The cell state CtC_t is updated by forgetting the previous state and adding the new candidate state.

5. Output Gate:

ot=σ(Wo·[ht−1,xt]+bo)

o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)

The output gate determines what the next hidden state hth_t will be.

6. Hidden State Output:

ht=ot·tanh(Ct)h_t = o_t \cdot \tanh(C_t)

The hidden state hth_t is generated by passing the cell state CtC_t through a tanh activation function and then applying the output gate.

LSTM Training Process:

a) Initialize Weights: Randomly initialize the weights and biases of the gates.

b) Forward Pass: Process the input sequence one step at a time and compute the hidden states and cell states using the gates.

c) Loss Calculation: Compute the loss (e.g., cross-entropy loss or MSE) based on the predicted output and the actual target.

d) Backpropagation Through Time (BPTT):

e) Compute gradients and backpropagate through time to adjust the weights and biases of the LSTM.

f) Repeat: Repeat the process over multiple epochs until the model converges.

RESULTS:

Figure 2. RNN-Accuracy

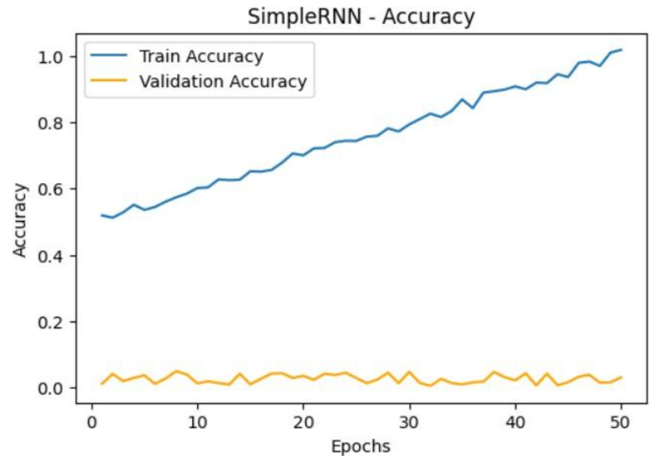


Figure 3. RNN-Loss

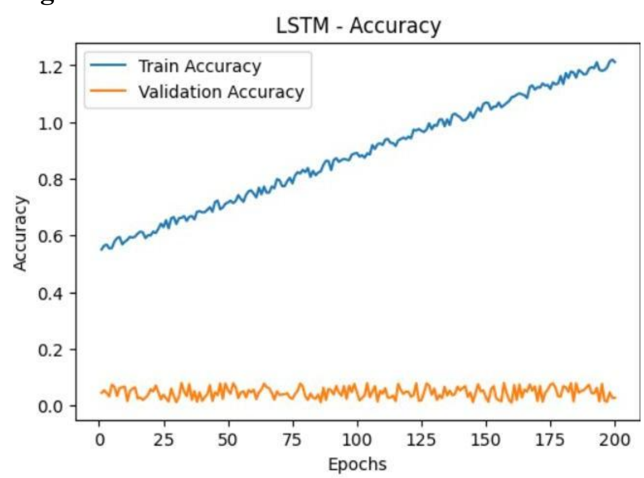


Figure 4. LSTM-Accuracy

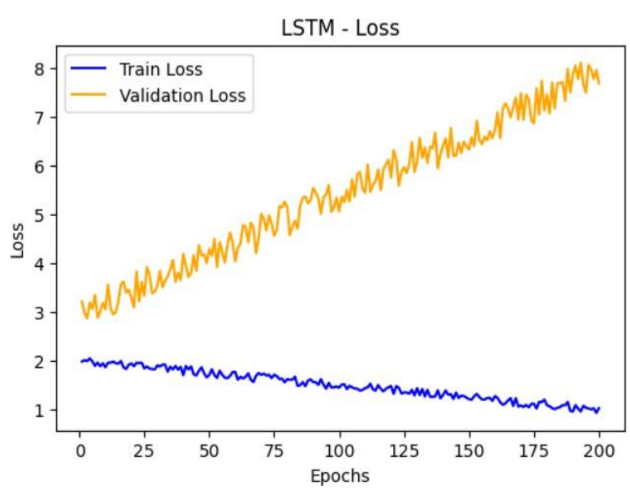
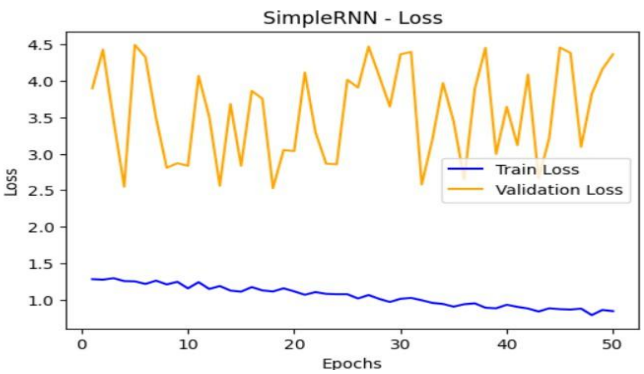


Figure 5. LSTM-Loss



VII. CONCLUSION:

In this project, we successfully implemented a RNN-based model using SimpleRNN and LSTM architectures for analyzing heartbeat sounds with the Librosa library. The model demonstrated a robust ability to classify different heartbeats based on the extracted audio features, showcasing the potential of deep learning techniques in biomedical signal processing. The integration with Django provided a seamless interface for users to upload audio files and receive real-time predictions, highlighting the practical applicability of this research in clinical settings.

VIII. FUTURE WORKS:

For future work, we aim to enhance the model's accuracy by exploring advanced techniques such as attention mechanisms and GRU (Gated Recurrent Unit) layers.Additionally, we plan to expand the dataset by incorporating diverse heartbeat sounds from different demographics and conditions to improve the model's generalization. Future iterations of the application could also include a user-friendly visualization of the audio features and prediction results, facilitating a deeper understanding of the analysis for healthcare professionals. Lastly, integrating this system with wearable devices could pave the way for real-time monitoring and early detection of cardiac abnormalities.

IX. REFERENCE:

[1] Bonow RO, Mann DL, Zipes DP, Libby P. Braunwald's heart disease e-book: A textbook of cardiovascular medicine. Philadelphia (PA): Elsevier Health Sciences; 2011.

[2] Etchells E, Bell C, Robb K. Does this patient have an abnormal systolic murmur? JAMA. 1997;277(7):564–571. [PubMed] [Google Scholar]

[3] Mangione S, Nieman LZ. Cardiac auscultatory skills of internal medicine and family practice trainees: A comparison of diagnostic proficiency. JAMA. 1997;278(9):717–722. [PubMed] [Google Scholar]

[4] Chen W, Sun Q, Chen X, Xie G, Wu H, Xu C. Deep learning methods for heart sounds classification: A systematic review. Entropy. 2021;23(6):667. [DOI] [PMC free article] [PubMed] [Google Scholar]

[5] Li S, Li F, Tang S, Xiong W. A review of computer-aided heart sound detection techniques. Biomed Res Int. 2020;2020(1):5846191. [DOI] [PMC free article] [PubMed] [Google Scholar]

[6] Sathyanarayanan S, Murthy S, Chitnis S. A comprehensive survey of analysis of heart sounds using machine learning techniques to detect heart diseases. J Popul Ther Clin Pharmacol. 2023;30(11):375–384. [Google Scholar]

[7] Ren Z, Chang Y, Nguyen TT, Tan Y, Qian K, Schuller BW. A comprehensive survey on heart sound analysis in the deep learning era. arXiv. 2023. https://doi.org/10.48550/arXiv.2301.09362

[8] Bentley P, Nordehn G, Coimbra M, Mannor S. The PASCAL classifying heart sounds challenge 2011 (CHSC2011)
results.
http://www.peterjbentley.com/heartchallenge/index.html

[9] Liu C, Springer D, Moody B, Silva I, Johnson A, Samieinasab M, Sameni R, Mark R, Clifford GD. Classification of heart sound recordings—The physionet computing in cardiology challenge 2016. PhysioNet. 2016. [Google Scholar]

[10] Oliveira J, Renna F, Costa PD, Nogueira M, Oliveira C, Ferreira C, Jorge A, Mattos S, Hatem T, Tavares T, et al. The CirCor DigiScope dataset: From murmur detection to murmur classification. IEEE J Biomed Health Inform. 2021;26(6):2524–2535. [DOI] [PMC free

[11] article] [PubMed] [Google Scholar]

[12] Yaseen, Son GY, Kwon S. Classification of heart sound signal using multiple features. Appl Sci. 2018;8(12):2344. [Google Scholar]

[13] Dong F, Qian K, Ren Z, Baird A, Li X, Dai Z, Dong B, Metze F, Yamamoto Y, Schuller BW.

© 2026 The Author(s). Published by IJCOPE Journal. Website: https://ijcope.org/

Machine listening for heart status monitoring: Introducing and benchmarking HSS—the Heart Sounds Shenzhen Corpus. IEEE J Biomed Health Inform.2019;24(7):2082–2092. [DOI] [PubMed] [Google

[14] Scholar]Kazemnejad A, Gordany P, Sameni R.EPHNOGRAM:Asimultaneous electrocardiogramand phonocardiogram database (version 1.0.0). PhysioNet. 2021. [DOI] [PubMed] [Google Scholar]

[15] Sameni R, Samieinasab M. Shiraz university fetal heart sounds database (version 1.0.1). PhysioNet. 2021. [Google Scholar].