SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I5P102

Volume 11 Issue 5, 18-26, May 2024

Original Article

Hybrid Whale Optimisation with Improved Pooling Based Recurrent Neural Network to Predict the Heart Murmur at Earlier Stage

M. Hemalatha[1], Zayaraz Godandapani[2]

[1,2]*Department of Computer Science, Pondicherry Technological University, Puducherry, India.*

[1]*Corresponding Author : mail.mhemalatha@gmail.com*

Received: 08 March 2024 Revised: 17 April 2024 Accepted: 05 May 2024 Published: 31 May 2024

Abstract - *Heart murmurs are abnormal sounds produced during cardiac cycles that can indicate underlying cardiovascular disorders. Early detection of heart murmurs is crucial for timely intervention and enhanced patient results. This paper proposed a hybrid method for early stage heart murmur prediction using a combination of the Whale Optimization Algorithm (WOA) and an Improved pooling-based Recurrent Neural Network (IRNN). The improved pooling mechanism enhances the RNN's ability to capture relevant features from the input data while reducing the computational complexity. Our proposed Model (WOA-IRNN) is able to learn the long-term reliance on the heart sound signals and identify the subtle patterns that are indicative of heart murmur. Experimental results demonstrate that the Whale Optimization with improved pooling-based RNN outperforms existing methods. We evaluated the WOA-IRNN method on a dataset of heart sound signals and realised a high accuracy of 98.9% in predicting heart murmur. The consequences illustrate that our system significantly progresses the prediction accuracy, demonstrating its potential for early stage heart murmur detection.*

Keywords - *Heart murmurs, Recurrent Neural Network, Whale Optimization Algorithm, Early detection, and classification.*

1. Introduction

Heart murmurs, abnormal sounds heard during cardiac cycles, can serve as crucial indicators of underlying cardiovascular conditions. Timely detection of heart murmurs is of paramount importance as it allows for early intervention and improved patient outcomes. The advancements in research have explored the potential of utilising computational methods to aid in the early stage prediction of heart murmurs [1]. A heart murmur is a common heart sound that several different factors and conditions, such as valve problems, congenital heart defects, and infections, can cause. Early detection of heart murmur can help to prevent serious complications, such as heart failure and stroke [2].

Traditional methods for heart murmur detection primarily rely on auscultation, where a healthcare provider listens to the sounds using a stethoscope. However, early-stage heart murmurs can be challenging to detect due to their subtle nature and the potential for other ambient noises to mask them [3]. As a result, there is a growing need for computational techniques that can assist in early-stage heart murmur prediction and facilitate timely intervention [4]. In recent years, machine learning and artificial intelligence have shown promise in various healthcare applications, including cardiac diagnosis. These techniques can analyse huge volumes of data and identify complex configurations that may be indicative of early-stage heart murmurs. By leveraging the power of machine learning algorithms, researchers aim to develop accurate and efficient models that can assist healthcare professionals in the early detection of heart murmurs [5].

Recurrent Neural Networks (RNNs) have shown remarkable success in various sequence modeling tasks, covering things like natural language processing and recognition of speech, along with things like time series analysis [6]. However, traditional RNN architectures have limitations in capturing long-term dependencies and efficiently extracting informative features from sequential data. To overcome these challenges, researchers have proposed various enhancements to RNNs, such as improved pooling mechanisms, to improve their performance and capabilities.

Pooling operations play a crucial role in feature extraction by summarising and condensing the information learned from the input sequences. Traditional pooling techniques, such as max pooling or average pooling, have been widely used in Convolutional Neural Networks (CNNs). However, directly applying these pooling methods to RNNs may not fully exploit the temporal dynamics and correlations present in sequential data [7]. These improved pooling techniques aim to capture and preserve the most salient features while discarding redundant or noisy information. By incorporating such pooling mechanisms, RNN models can more effectively represent and extract relevant features from sequential data, leading to improved performance in various tasks [8].

The WOA mimics the hunting behavior of humpback whales by defining three main operators: encircling, spiral, and bubble-net. These operators guide the search process for the purpose of successfully exploring and exploiting the search space [9]. The algorithm is designed to locate the best possible answers for the many different optimisation challenges, including numerical optimisation, engineering design, and machine learning. The WOA's unique combination of exploration and exploitation strategies, modelled after the activities of humpback whales in the wild, helps to strike a balance between efficient exploration of the search space and manipulation of promising regions [10]. The algorithm has demonstrated competitive performance and has been effectively applied to resolve a wide assortment of optimisation difficulties.

Despite the advancements in medical technology, early detection of heart murmurs remains a challenging task due to the complexity of heart sound signals and the presence of various confounding factors. Existing methods often rely on simplistic feature extraction techniques or fail to capture the intricate patterns present in heart sound data adequately. Furthermore, many approaches suffer from high computational complexity, making them impractical for real-time or large-scale deployment.

Therefore, there is a pressing need for innovative techniques that can effectively extract relevant features from heart sound signals while addressing the computational challenges associated with real-world applications. Additionally, achieving higher accuracy rates in heart murmur prediction is crucial for ensuring timely intervention and improving patient outcomes."

With this problem statement, the study aims to address the identified research gap by proposing the hybrid method (WOA-IRNN) for early stage heart murmur prediction, which overcomes the limitations of existing approaches and demonstrates significant improvements in prediction accuracy.

The rest of the paper is as follows: Section 2 provides a comprehensive review of related studies and techniques in heart murmur prediction. Section 3 describes the proposed Whale Optimisation with improved pooling-based Recurrent Neural Network in detail. Section 4 presents the results obtained from the experiments conducted to evaluate the performance of the Whale Optimization with improved pooling-based RNN for early stage heart murmur prediction. Section 5 summarises the key findings of this research.

2. Related Works

Producing various segmentations of the signal is the goal of this method, which does this by applying many Markov models, each of which makes a unique set of assumptions on the possibility of a murmur [11]. After that, in order to develop a murmur categorisation as well as a robust segmentation, we evaluate the confidence of the output produced by each Markov model. The aim of this paper [12] is to advance a DNN and evaluate its capabilities in relation to those of other modern Machine Learning (ML) approaches. During the testing, the normal of the linked characteristics was used to substitute for any values in the database that were absent.

It is usual practice to use a trained neural network classifier for the purpose of predicting cardiac problems by classifying heart sound signals, which are also referred to as PCG signals [13]. On the other hand, there is still some debate on which training optimisation method is the most effective for solving problems of this kind, including classification. This study [14], suggested an automated methodology established on heart sounds using CRNN. This approach offers a novel, non-intrusive and simple method for heart failure typing.

A streptococcal infection of the throat may lead to rheumatic heart disease, which is an illness of the heart that can lead to organ damage, permanent valve destruction, and heart failure [15]. The illness might be preceded by a condition known as acute rheumatic fever. This research [16] makes use of methods to provide reliable heart murmur identification using phonocardiogram (PCG) recordings. Key pre-processing procedures include normalisation of the PCG segments, study of the PCG and segmentation.

The stethoscope readings are analysed in this study with the purpose of diagnosing a number of conditions that may lead to heart failure. The fundamental objective of this study [17] is to locate and organise the data relating to the many types of heart sounds, which have been grouped into the broad categories of S1 through S4. Detecting both of the initial heart sounds, often referred to as S1 as well as S2 accordingly, is generally the initial phase in the method of detecting any form of cardiac irregularities, and it is also the stage that is considered to be the most significant. This article [18] provides a framework for an optimal 1-dimensional CNN algorithm which makes use of edge computing. The CNN is perfect since it only has one dimension.

The diagnosis of cardiac disease in its earliest stages is a primary objective in paediatric cardiology. The development of a low-cost screening device that may aid in the discrimination between benign and pathological cardiac murmurs has met with very modest success thus far. The purpose of this research [19] was to successfully teach an ANN to differentiate between normal and abnormal murmurs in a patient's heart. Young doctors and those with little clinical experience sometimes struggle when it comes to making an accurate diagnosis of aberrant heart sound patterns. In the course of this research [20], wants to create an innovative algorithm that is capable of the automated recognition of systolic murmurs in patients who have ventricular septal defects.

This work [21] intended to produce a stethoscope that would be driven by machine learning and termed SmartScope. Its purpose would be to assist medical professionals in the quick analysis, proof, and enhancement of cardiopulmonary auscultation. Because of its capacity to unearth hidden links and patterns in healthcare data, CNNs have proven to be incredibly useful in the development of systems that assist patients' health [22]. The prediction of heart problems by analysing cardiac abnormalities is one of the uses of such systems that is considered to be among the most significant and valuable of its kind.

The non-invasive, hassle-free, and reasonably inexpensive method of heart sound auscultation continues to be the method of choice when it comes to deciding whether or not a patient has cardiovascular disease [23]. This article [24] offers a technique for diagnosing abnormal heart sounds based on a unique Dense Feature Selection CNN architecture. The authors of this article developed the approach. CNN has seen extensive use in the processing of images. The approach of analysing heart sounds with the use of CNN has been researched in this particular portion of the work [25].

Our major objective is to demonstrate that the proposed technique can outperform current approaches and correctly forecast heart murmurs at an earlier stage. This will be accomplished by showing that it can overcome the other existing methods. Medical professionals can quickly intervene, provide patients with the proper therapy, and maybe stop the course of cardiovascular problems if they are able to achieve early identification.

3. Proposed Model

In this section, we proposed a Hybrid Whale Optimisation algorithm with an improved pooling-based Recurrent Neural Network for early stage heart murmur prediction. Collect auscultatory recordings of heart sounds, including normal and abnormal (murmur) cases. PCG signal is given as input, and the pre-processed signals into CNN as feature extraction. Utilise the WOA as an optimisation algorithm to optimise the hyperparameters of the Improved Pooling-Based Recurrent Neural Network.

We proposed attention mechanisms to improve the pooling operation, allowing the network to focus on relevant features for heart murmur prediction. An overall system model is shown in Figure 1.

The architecture of the RNN, including the improved pooling mechanism enhances its feature extraction capabilities as explained with the Whale Optimisation Algorithm and outlines the parameters of the RNN.

3.1. Feature Extraction Using CNN Model

CNNs have computer vision, bringing about a revolution in the discipline by exhibiting exceptional presentation in several image-related tasks, including feature extraction. CNN models are designed to automatically discover and retrieve hierarchical structure representations of visual features from input images, enabling them to capture spatial and local patterns effectively, as shown in Figure 2.

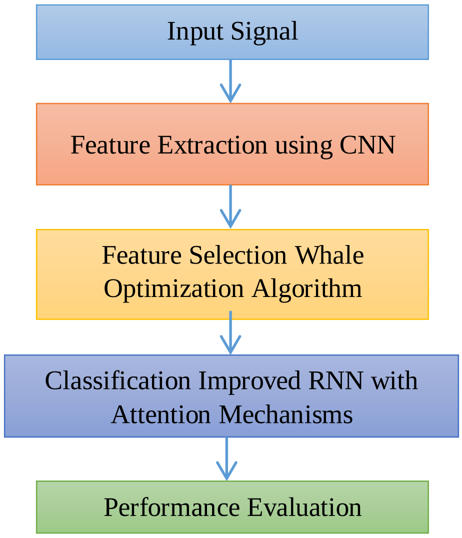
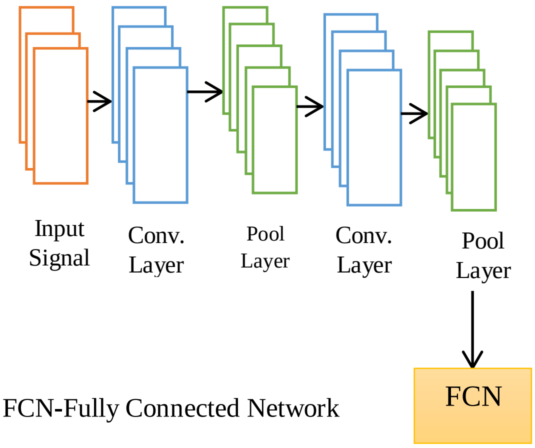


Fig. 1 Overall system model



* FCN-Fully Connected Network

* Conv-Convolution, Pool-Pooling

Fig. 2 CNN on the feature extraction

CNNs employ convolutional layers that perform convolutions on the input data using learnable filters or kernels. By moving across the input image while calculating element-wise multipliers and summations, these filters were developed to recognise certain visual patterns, such as edges, which are textures or forms.

The output feature map at position (a, b) for the n-th filter can be calculated as:

$$S[a,b,n] = activation(\Sigma\Sigma\Sigma\ P[i,j,c] * W[n,i,j,c] + f[n])$$

Here,

P[i,j, c] represents the input image at position (p, q) in channel c.

W[k, i, j, c] denotes the weight of the n-th filter at position (p, q) in channel c.

f[n] is the bias term for the n-th filter.

$activation$ (.) computes the sum of the activation function.

After the convolutional layer, pooling operations are often performed to downsample the feature maps and reduce their spatial dimensions.

The output feature map at position (a,b) for the n-th channel can be computed as:

$$P[a,b,n] = max(S[a * O: a * O + P, b * O: b * O + P, n])$$

Here,

$S[a * O: a * O + P, b * O: b * O + P, n]$ denotes the region of the input feature map that the pooling operation is applied to. $max(.)$ returns the maximum value within the pooling region.

Pseudocode for Feature Extraction

```
def extract_features(signals, model):
//Extracts features from a set of signals using a CNN model.
  Args:
    signals: A list of signals.
    model: A CNN model.
  Returns:
  //  A list of feature vectors.
    feature_vectors = []
  for signal in signals:
    feature_vector = model.predict(signal)
    feature_vectors.append(feature_vector)
  return feature_vectors
def main():
//Extracts features from a set of signals and saves them to a file.
  signals = load_signals()
  model = load_model()
  feature_vectors = extract_features(signals, model)
  save_feature_vectors(feature_vectors)
```

From an algorithm, pre-processed the PCG murmur samples and split them into training and testing datasets. The model architecture should be defined based on your specific requirements, such as the number of convolutional layers, their filters and kernel sizes, pooling layers, and fully connected layers.

3.2. Feature Selection Using WOA

In the context of feature selection, the WOA can be applied to find an optimal subset of features that maximises the performance of a DL model while reducing computational costs and avoiding overfitting. Feature selection is a precarious step in data pre-processing, as it helps to identify the most relevant and informative features for the given task. The WOA algorithm consists of three main operators:

3.2.1. Encircling Prey

This operator simulates the behavior of humpback whales encircling their prey. In this operator, a whale randomly chooses one of its neighbors and moves towards it. Update the positions of other whales using the following equation:

$$new\_position = best\_position - A * distance\_to\_best * (1 - 2 * random\_value)$$

Here, A is the encircling coefficient that decreases linearly from 2 to 0 over iterations. distance_to_best is the Euclidean distance between the current whale position and the best position found. random_value is a random number between 0 and 1.

3.2.2. Breaching

This operator pretends the behavior of humpback whales breaching out of the water. In this operator, a whale randomly generates a new solution and moves towards it.

Update the positions of a random subset of whales by performing a movement towards the average position of the other whales. The new positions can be updated using the following equation:

$$new\_position = average\_position - B * (1 - 2 * random\_value)$$

Here, B is a constant controlling the movement intensity. random_value is a random number between 0 and 1.

Pseudocode for Feature Selection Using WOA

```
population_size = 30
max_iterations = 100
search_space = num_features  # Number of features in the dataset
population = initialize_population(population_size, search_space)
fitness_values = evaluate_fitness(population)
best_solution = population[np.argmax(fitness_values)]
best_fitness = np.max(fitness_values)
for iteration in range(max_iterations):
    a = 2 - iteration * (2 / max_iterations)  # Update the encircling coefficient
    for i in range(population_size):
        distance_to_best = np.abs(best_solution - population[i])
        new_position = best_solution - a * distance_to_best
        new_fitness = evaluate_fitness(new_position)
        if new_fitness > best_fitness:
            best_solution = new_position
            best_fitness = new_fitness
selected_features = best_solution
```

3.2.3. Bubble-Net Feeding

This operator simulates the behavior of humpback whales using bubble-net feeding to catch their prey. In this operator, a whale randomly generates a new solution and moves towards it, but with a probability of being attracted back to its original solution. Update the positions of the remaining whales using a random search strategy.

The new positions can be updated using the following equation:

$$new\_position = prey\_position - distance\_to\_prey * (1 - 2 * random\_value)$$

Here, prey_position is a random whale position. distance_to_prey is the Euclidean distance between the current whale position and the prey position. random_value is a random number between 0 and 1.

3.3. Improved Pooling - Based Recurrent Neural Network for Classification Model

The improved pooling-based RNN is a modification of the traditional RNN architecture that incorporates a pooling mechanism to enhance feature extraction and computational efficiency. This modification aims to address some of the limitations of standard RNNs, such as the inability to effectively capture long-term dependencies and the computational complexity associated with processing sequential data. In the improved pooling-based RNN, the pooling mechanism is introduced to summarise the hidden states of the recurrent units over a specific time interval. This pooling operation reduces the dimensionality of the hidden states while preserving important information, allowing the network to focus on the most relevant features for prediction.

One approach to improving pooling-based RNNs for classification is the introduction of hierarchical pooling. In this technique, the RNN outputs are first grouped into segments, such as time steps or chunks. Then, pooling operations are performed within each segment to capture the most relevant information.

The pooled outputs from each segment are further aggregated using another pooling operation to obtain a fixed-length representation of the entire sequence. This hierarchical pooling allows the Model to capture both local and global dependencies in the input sequence.

Attention mechanisms can be integrated with pooling-based RNNs to enhance their classification performance further. Attention mechanisms enable the Model to assign different weights or varying degrees of significance to various components of the input sequence. Because of the incorporation of attention into the pooling-based RNN, the Model can selectively attend to relevant information and suppress irrelevant or noisy inputs, leading to improved classification accuracy. An overall proposed architecture is shown in Figure 3.

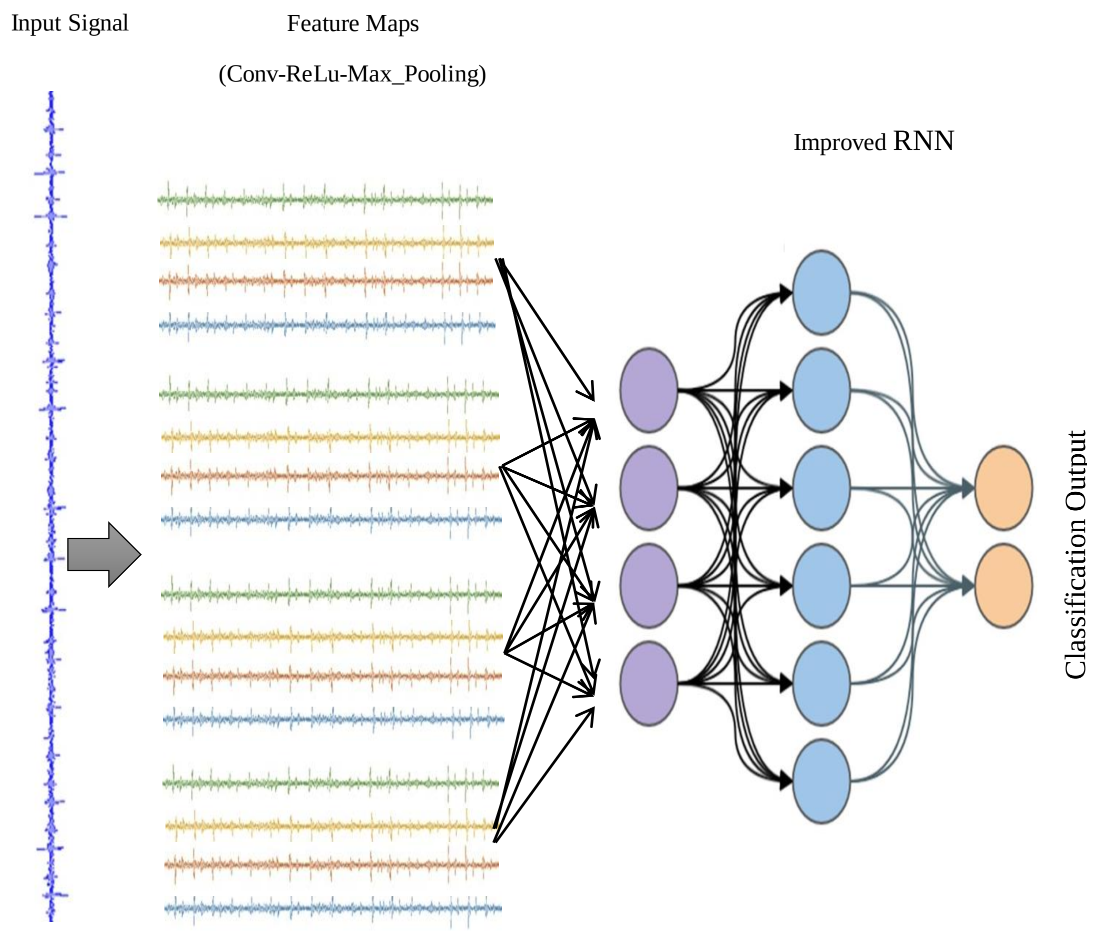


Fig. 3 The proposed Recurrent Neural Network

The pooling operation can be performed at different levels of the RNN architecture. For example, it can be applied at each time step within a recurrent layer or at the end of the recurrent layer to summarise the entire sequence. This flexibility allows researchers to explore different pooling strategies based on the specific requirements of the task at hand. Pooling contributes to computational efficiency by decreasing the number of variables and computations required in the RNN model. This can be particularly advantageous when dealing with large-scale or real-time applications where computational resources are limited.

3.3.1. Mathematical Equation

The mathematical equation for an Improved pooling-based Recurrent Neural Network (IRNN) for a classification model depends on hierarchical pooling and attention mechanisms.

Let us assume an input sequence of length T, represented as x = (x1, x2, ..., xT), where xt denotes time. The goal is to classify the sequence into one of the K classes.

3.3.2. RNN Computation

The RNN computes hidden states h = (h1, h2, ..., hT) using recurrent equations, typically based on LSTM or GRU units. For example, an LSTM update equation can be defined as:

$$ht = LSTM(ht-1, xt)$$

3.3.3. Hierarchical Pooling

Hierarchical pooling involves grouping the hidden states into segments and applying pooling operations. Let us assume we divide the sequence into P segments, and within each segment, we apply a pooling operation.

For each segment p, we apply a pooling operation, such as max pooling or average pooling, to reduce the segment's hidden states to a fixed-length representation. This can be represented as:

$$hp = Pooling(h[p\_start : p\_end])$$

Where hp represents the pooled representation of segment p, and [p_start: p_end] denotes the range of hidden states corresponding to segment p.

To obtain a fixed-length representation of the entire sequence, we further apply a pooling operation to the pooled representations of all segments. This can be represented as:

$$h\_agg = Pooling(hp1, hp2, \ldots, hpP)$$

Where h_agg represents the aggregated representation of the entire sequence.

3.3.4. Attention Model

The Model's attention processes enable it to concentrate on important bits of the input stream. We compute attention weights using the hidden states and apply them to the pooled representations.

3.3.5. Attention Weights

Attention weights α = (α1, α2, ..., αP) are computed for each segment based on the hidden states and a learnable weight matrix W:

$$\alpha p = softmax(W * hp)$$

Where αp represents the attention weight for segment p.

3.3.6. Weighted Pooled Representation

To produce the final representation, calculate a weighted average of all the pooled models using attention weights:

$$h\_final = \sum(\alpha p * hp)$$

Where h_final represents the final representation of the sequence.

3.3.7. Classification Layer

Finally, the classification layer takes the final representation h_final and produces the output probabilities for each class using softmax or other activation functions.

Pseudocode for IRNN

```
def improved_pooling_based_rnn_classification_model(input_data):
rnn = RecurrentNeuralNetwork(input_data.shape[1])    # Initialize the RNN model.
  rnn_output = rnn.forward(input_data)         # Feed the input data to the RNN.
  pooled_output = improved_pooling(rnn_output)       # Apply the improved pooling layer.
weighted_average = weighted_average(pooled_output, rnn_output)  # Calculate the weighted average of the output units
pooled_output = weighted_average
  Classification = softmax(pooled_output)  # Classify the output using a softmax layer.
  return classification
```

The RNN model is initialised with the number of structures in the input data. The input data is fed to the RNN model. The improved pooling layer is applied to the RNN output. The weighted average of the output units is calculated by weighting each output unit by its corresponding temporal dependency. The output of the improved pooling layer is classified using a softmax layer. The softmax layer outputs a probability distribution over the possible classes.

4. Results and Discussions

4.1. Dataset Description

The dataset contains 3126 Phonocardiogram (PCG) recordings with simultaneously recorded Electrocardiograms (ECG), recorded for 10-60 seconds. It is common practice to pay attention to the murmurs of the heart in one of these four regions, which are called based on the positions at which the heart valves may be heard the clearest:

- Aortic area - located near the centre of the second intercostal gap to the right.
- Pulmonic area - across the left sternal boundary, in the subsequent intercostal gap.
- Tricuspid area - across the left sternal border, in the 4th intercostal gap.
- Mitral area - near the top of the heart, at the fifth intercostal gap, on the line that runs through the middle of the clavicle.

Table 1. Properties of the data sets used in this study. FS: sampling frequency in HZ

Dataset	FS	Total Samples	Distribution
CinC2016	2000	3240	665 abnormal, 2575 normal

As seen in Table 1, the data sets are composed of records that were sampled at varying frequencies during their collection. In order to get rid of this issue during the pre-processing step, all of the data were resampled at a frequency of one thousand hertz. Additionally, PCG signals were normalised to fall within the range of [-1] to [1].

4.2. Performance Metrics

Accuracy, precision, recall, F1 score, and weighted accuracy are all metrics used to evaluate the performance of learning models.

- The percentage of all forecasts that turned out to be accurate is the accuracy. The method for calculating accuracy is:

$$Accuracy = \frac{TP + TN}{Total\ Predictions}$$

- Precision is the proportion of positive predictions that things were in point of fact, positive. To determine it, the sum of the numbers of genuine positives and false positives and divided by the overall amount of genuine positives. This will give you the percentage of false positives.

Precision = TP / (TP + FP)

- Recall is the percentage of real positives that can be accurately forecasted as positives. To determine it, divide the number of true positives by the total number of true positives and false negatives. This will give you the accuracy rate.

Recall = TP / (TP + FN)

- The F1 score is a weighted harmonic mean of precision and recall. It is calculated by

F1_Score = 2*(Precision*Recall)/(Precision+Recall)

- Weighted accuracy is a measure of accuracy that takes into account the imbalance of classes in a dataset. It is calculated by weighting the accuracy of each class by its proportion in the dataset.

Weighted_accuracy = sum(class_accuracy * class_proportion)

Table 2. Matrix weighed accuracy

		Prediction		
		Present [1]	Unknown [-1]	Absent [0]
Murmur	Present [1]	M [1] [1]	M [1] [−1]	M[1] [0]
	Unknown [-1]	M[−1] [1]	M[−1] [−1]	M[−1] [0]
	Absent[0]	M[0] [1]	M[0] [−1]	M[0] [0]

The confusion matrix helps to assess the performance of a classification model by providing insights into the Model's ability to identify positive and negative instances correctly shown in Figure 4.

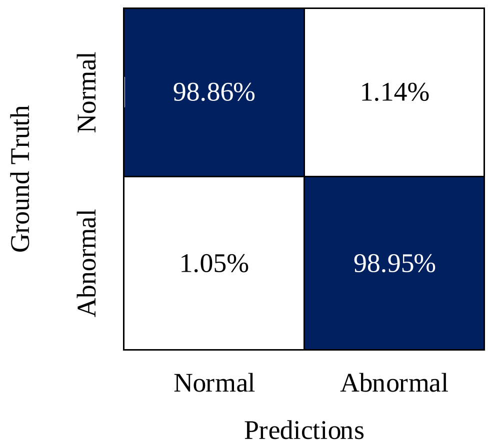


Fig. 4 Confusion matrix

An accuracy curve, also known as an accuracy plot or learning curve, is a graphical representation in Figure 5 shows how the accuracy of a learning model changes as a function of different factors.

Table 3. Performance evaluation

Model	Accuracy	Precision	Recall	F1-Score
CNN	95.6	96.56	93.21	92.45
RNN	96	94.2	93.56	93.12
LSTM	96.45	94.23	94.45	93.25
CNN + Bi-LSTM	98.3	96.12	96.24	94.3
Proposed (IRNN-WOA)	98.9	97.26	97.86	97.61

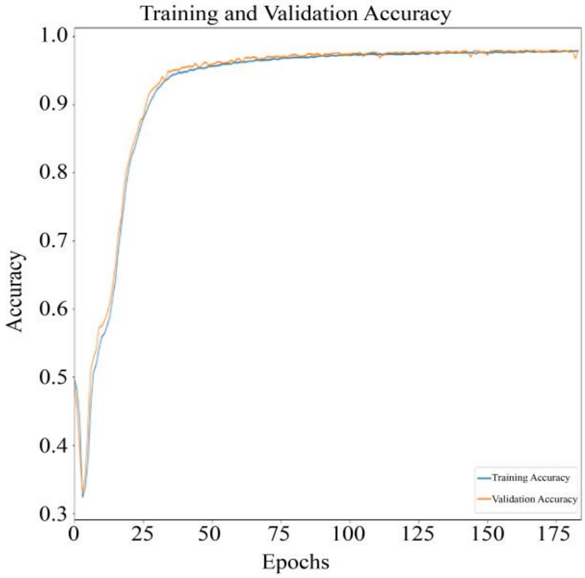


Fig. 5 Accuracy curve

The ROC curve is a plot of the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The TPR is the proportion of positive instances that are correctly classified, while the FPR is the proportion of negative instances that are incorrectly classified is shown in Figure 6.

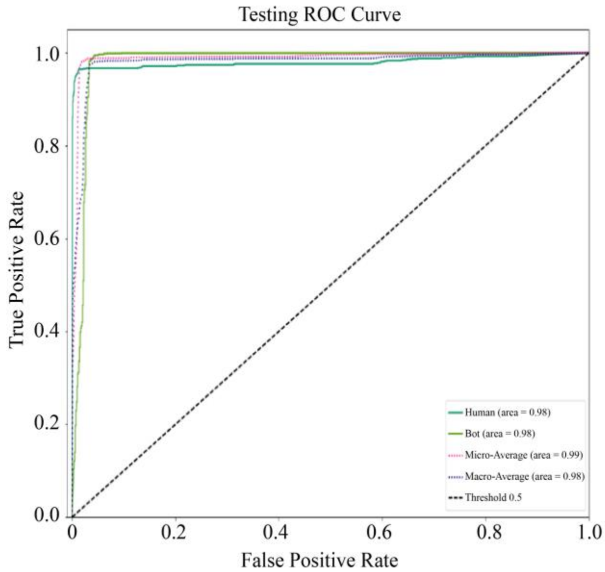


Fig. 6 ROC curve

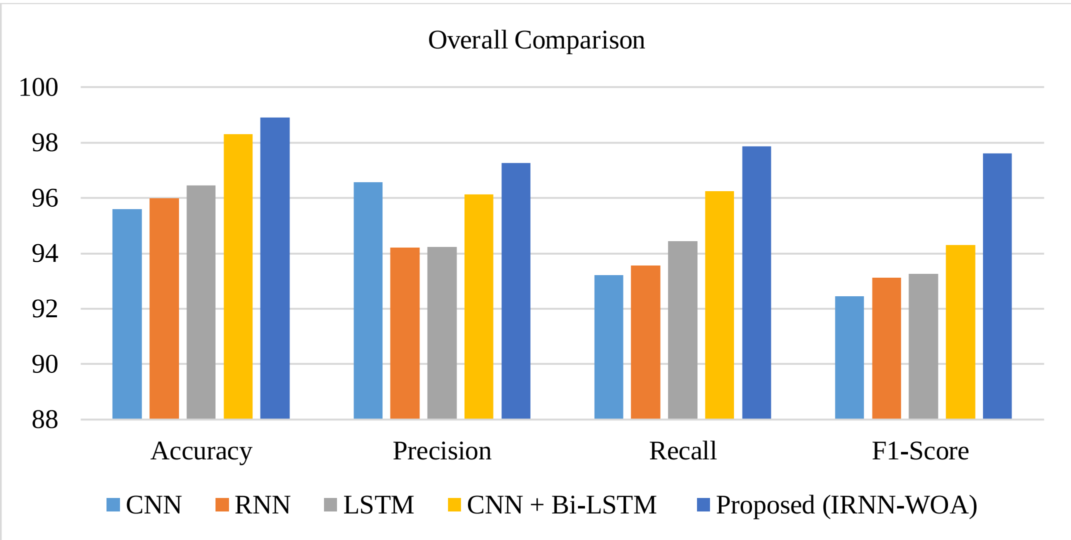


Fig. 7 Overall comparison of proposed model

The results represented in Figure 7 show that our proposed Model achieves better accuracy in the early prediction of Heart Murmurs than other existing models.

5. Conclusion

By combining the optimisation power of the whale optimisation algorithm with the enhanced feature capturing capabilities of improved pooling-based RNNs, this method aims to improve the accuracy and timeliness of heart murmur detection. The WOA mimics the social behavior of humpback whales, enabling efficient exploration and exploitation of the search space to find optimal parameter values. By leveraging this optimisation technique, the Model can effectively learn and adapt to the complex patterns and characteristics of heart murmurs. The improved pooling-based RNN architecture incorporates advanced pooling techniques such as hierarchical pooling, adaptive pooling, and attention mechanisms.

These techniques enable the Model to capture local and global dependencies within the heart murmur sequences, focus on the most relevant segments, and assign different weights to different parts of the input sequence based on their importance. By effectively pooling and attending to the salient features, the Model enhances its predictive capability for heart murmur detection.

References

[1] Yujia Xu et al., “Hierarchical Multi-Scale Convolutional Network for Murmurs Detection on PCG Signals,” *2022 Computing in Cardiology (CinC)*, Tampere, Finland, pp. 1-4, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Matthew A. Reyna et al., "Heart Murmur Detection from Phonocardiogram Recordings: The George B. Moody PhysioNet Challenge 2022," *PLOS Digital Health*, vol. 2, no. 9, pp. 1-22, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Maurice Rohr et al., "Two-Stage Multitask-Learner for PCG Murmur Location Detection," *2022 Computing in Cardiology (CinC)*, Tampere, Finland, pp. 1-4, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Jiaming Wang et al., "Intelligent Diagnosis of Heart Murmurs in Children with Congenital Heart Disease," *Journal of Healthcare Engineering*, vol. 2020, pp. 1-9, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Shahnawaz Alam, Rohan Banerjee, and Soma Bandyopadhyay, "Murmur Detection Using Parallel Recurrent & Convolutional Neural Networks," *arXiv*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Yunbo Wang et al., "PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2208-2225, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Omer Deperlioglu et al., "Diagnosis of Heart Diseases by a Secure Internet of Health Things System Based on Autoencoder Deep Neural Network," *Computer Communications*, vol. 162, pp. 31-50, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Mohanad Alkhodari, and Luay Fraiwan, "Convolutional and Recurrent Neural Networks for the Detection of Valvular Heart Diseases in Phonocardiogram Recordings," *Computer Methods and Programs in Biomedicine*, vol. 200, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Sanjoy Chakraborty et al., "A Novel Improved Whale Optimization Algorithm to Solve Numerical Optimization and Real-World Applications," *Artificial Intelligence Review*, vol. 55, pp. 4605-4716, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Rohit Kundu et al., "AltWOA: Altruistic Whale Optimization Algorithm for Feature Selection on Microarray Datasets," *Computers in Biology and Medicine*, vol. 144, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Andrew McDonald, Mark J.F. Gales, and Anurag Agarwal, "Detection of Heart Murmurs in Phonocardiograms with Parallel Hidden Semi-Markov Models," *2022 Computing in Cardiology (CinC)*, Tampere, Finland, pp. 1-4, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] F. Mokeddem, F. Meziani, and S.M. Debbal, "Study of Murmurs and their Impact on the Heart Variability," *International Journal of Medical Engineering and Informatics*, vol. 12, no. 3, pp. 291-301, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Vijendra Singh, Vijayan K. Asari, and Rajkumar Rajasekaran, "A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease," *Diagnostics*, vol. 12, no. 1, pp. 1-22, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Sara Khaled, "Analysis of Training Optimization Algorithms in the NARX Neural Network for Classification of Heart Sound Signals," *International Journal of Scientific and Engineering Research*, vol. 13, no. 2, pp. 382-390, 2022. [Google Scholar] [Publisher Link]

[15] Hui Wang et al., "An Automatic Approach for Heart Failure Typing Based on Heart Sounds and Convolutional Recurrent Neural Networks," *Physical and Engineering Sciences in Medicine*, vol. 45, no. 2, pp. 475-485, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] K. Antony Kumar, and M.J. Carmel Mary Belinda, "A Multi-Layer Acoustic Neural Network-Based Intelligent Early Diagnosis System for Rheumatic Heart Disease," *International Journal of Image and Graphics*, vol. 24, no. 1, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ahmed Patwa, Muhammad Mahboob Ur Rahman, and Tareq Y. Al-Naffouri, "Heart Murmur and Abnormal PCG Detection via Wavelet Scattering Transform & a 1D-CNN," *arXiv*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Yasser Zeinali, and Seyed Taghi Akhavan Niaki, "Heart Sound Classification Using Signal Processing and Machine Learning Algorithms," *Machine Learning with Applications*, vol. 7, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Venkatesh Vakamullu et al., "Convolutional Neural Network Based Heart Sounds Recognition on Edge Computing Platform," *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Ottawa, ON, Canada, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Curt G. DeGroff et al., "Artificial Neural Network–Based Method of Screening Heart Murmurs In Children," *Circulation*, vol. 103, no. 22, pp. 2711-2716, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[21] Jou-Kou Wang et al., "Automatic Recognition of Murmurs of Ventricular Septal Defect Using Convolutional Recurrent Neural Networks with Temporal Attentive Pooling," *Scientific Reports*, vol. 10, no. 1, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Aadhav Prabu, "SmartScope: An AI-Powered Digital Auscultation Device to Detect Cardiopulmonary Diseases," *TechRxiv, Authorea Preprint*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Ali A. Samir et al., "Evolutionary Algorithm-Based Convolutional Neural Network for Predicting Heart Diseases," *Computers & Industrial Engineering*, vol. 161, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Xin Zhou et al., "A Novel 1-D Densely Connected Feature Selection Convolutional Neural Network for Heart Sounds Classification," *Annals of Translational Medicine*, vol. 9, no. 24, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Yi He et al., "Research on Segmentation and Classification of Heart Sound Signals Based on Deep Learning," *Applied Sciences*, vol. 11, no. 2, pp. 1-15, 2021. [CrossRef] [Google Scholar] [Publisher Link]