H-Infinity Filter Enhanced CNN-LSTM for Arrhythmia Detection from Heart Sound Recordings

Rohith Shinoj Kumar, Rushdeep Dinda, Aditya Tyagi, Annappa B., Naveen Kumar M. R.
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal, Karnataka, India 575025
rohith.211cs245@nitk.edu.in, rdinda.211cs246@nitk.edu.in, adityatyagi.211cs101@nitk.edu.in,
annappa@ieee.org, naveenkumarmr.227cs003@nitk.edu.in

***Abstract*—Early detection of heart arrhythmia can prevent severe future complications in cardiac patients. While manual diagnosis still remains the clinical standard, it relies heavily on visual interpretation and is inherently subjective. In recent years, deep learning has emerged as a powerful tool to automate arrhythmia detection, offering improved accuracy, consistency, and efficiency. Several variants of convolutional and recurrent neural network architectures have been widely explored to capture spatial and temporal patterns in physiological signals. However, despite these advancements, current models often struggle to generalize well in real-world scenarios, especially when dealing with small or noisy datasets, which are common challenges in biomedical applications. In this paper, a novel CNN-H-Infinity-LSTM architecture is proposed to identify arrhythmic heart signals from heart sound recordings. This architecture introduces trainable parameters inspired by the H-Infinity filter from control theory, enhancing robustness and generalization. Extensive experimentation on the PhysioNet CinC Challenge 2016 dataset, a public benchmark of heart audio recordings, demonstrates that the proposed model achieves stable convergence and outperforms existing benchmarks, with a test accuracy of 99.42% and an F1 score of 98.85%.**

***Index Terms*—Arrhythmia detection, Deep Learning, H-infinity filter, CNN-LSTM, Phonocardiogram, Biomedical signal processing, Medical AI**

I. Introduction

Cardiovascular diseases remain the foremost contributor to global mortality, claiming nearly 18 million lives each year [1]. Furthermore, the number of deaths due to heart disease has risen faster than that of any other cause worldwide. Heart arrhythmias are irregularities in the heartbeat caused by disrupted electrical signals, leading to rhythms that are too fast (tachycardia), too slow (bradycardia), or erratic. If not identified early, persistent arrhythmias can weaken the heart muscle over time, making it less effective at pumping blood. This increases the chances of the patient suffering from strokes, heart failure, and cardiac arrest. Atrial Fibrillation (AF), for instance, impacts over 2.3 million individuals in the United States alone [2]. Early detection and management are crucial to preventing these complications.

A Phonocardiogram (PCG) is a non-invasive diagnostic tool that graphically represents heart sounds and murmurs, captured using a sensor placed on the chest. Figure 1 illustrates the 4 cardiac phases from an ECG and their corresponding interpretation from a standard heart audio sample of a healthy person. In this work, the focus is on the classification of

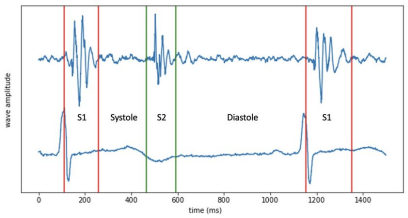


Fig. 1: Waveform representation of the cardiac cycle phases

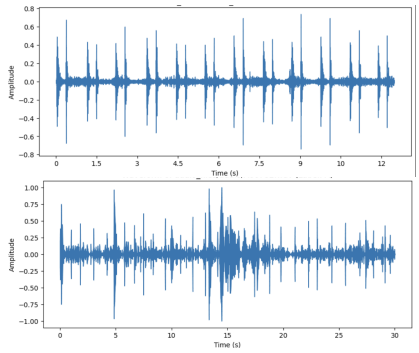


Fig. 2: Healthy (Top) and Arrhythmic (Bottom) Waveforms

cardiac arrhythmias from audio recordings of the cardiac rhythms based on deep learning approaches.

Heart sound classification has seen substantial advances in recent years, especially through the fusion of spectrogram representations and deep learning-based techniques. Nilanon et al. [3] were among the first to propose the use of spectrograms and CNNs for the classification of heart sounds, showing that time-frequency representations can greatly aid in enhanced classification.

Tsai et al. [4] presented a capsule network framework that undertakes the processing of spectrograms by convolutional capsule layers through dynamic routing to enable more hierarchical feature encoding. To tackle class imbalance, Li et al. [5] made use of the weighted loss and further enhanced the input by concatenating the Mel spectrogram with the Mel-Frequency Cepstral Coefficient (MFCC) features to better discriminate normal versus pathological instances.

Singh-Miller et al. [6] pursued a hybrid approach by applying principal component analysis (PCA) and k-means to

extract features that model the activity in different frequency bands, followed by random forest training of these features. Similarly, Vernekar et al. [7] extracted a mix of statistical and frequency-domain features, enhanced with Markov-chain analysis, to train a neural network. Chen et al. [8] devised a CNN-LSTM architecture that takes segmented raw audio as an input to instill both spatial and temporal patterns onto a signal. Deng et al. [9] proposed a convolutional recurrent neural networks (CRNN) based framework on enhanced MFCC features, experimenting with noise-ridden datasets. H_∞ filters are used in a speech enhancement method presented by [10], which highlights its potential in eliminating the need for prior knowledge of noise statistics, unlike conventional Wiener and Kalman filtering techniques.

In this work, training and evaluation was performed on recordings from the PhysioNet Computing in Cardiology Challenge (CinC) Arrhythmia Detection dataset [11], a public benchmark of audio recordings annotated for various cardiac conditions. The dataset contains approximately 8500 heart sound recordings at a sampling rate of 2000 Hz, thus containing rich acoustic details necessary for sound diagnostic purposes. However, some of these files are not annotated and such files have been filtered out for the purpose of this work, reducing the effective size of the dataset to roughly 6000 audio files. Figure 2 illustrates the waveforms of a healthy and arrhythmic sample from the dataset. However, while the dataset is a vast collection of high-quality audio recordings, working with the CinC dataset poses some very serious issues.

- Pronounced class imbalance: The data show a notable skew, with roughly 87% (5154 samples) of the recordings classified as normal, and a minority as abnormal heart sounds (771 samples).
- Variable recording length: The length of the recordings of the heart sounds varies between a few seconds to over one minute.
- The existence of noise and artifacts in many real heart sound recordings obscure the underlying cardiac signals.

In summary, this work makes two key contributions. Firstly, a novel deep learning architecture, the CNN-H_∞-LSTM, is proposed which replaces the classic forget gate and cell state update equations of an LSTM unit with trainable parameters inspired by the H_∞ filter from control theory. This modification draws on the H_∞ filter's well-established ability to minimize worst-case estimation errors under unknown noise, aiming for better generalization on small and noisy datasets like the one used in this study [12]. Secondly, it introduces a training optimization method called Stochastic Adaptive Probe Thresholding (SAPT), coupled with a custom loss function designed to address the issue of class imbalance. The remaining sections of the paper are organized as follows: Section II outlines the proposed methodology, while the experimental setup is illustrated in III. The experimental results of the proposed architecture, along with an extensive comparison with prior benchmark models, are presented in Section IV, followed by conclusion and the scope of future work in V.

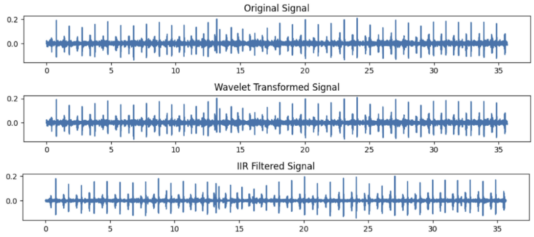


Fig. 3: Effect of wavelet and IIR Filter on the original signal

II. Proposed Methodology

This section discusses in detail the proposed CNN-H_∞-LSTM architecture trained to identify arrhythmia from a variable-length audio sample of a heart rhythm. This includes noise suppression of the audio sample, generation of Mel spectrograms from the transformed audio waveform and the subsequent training of the proposed architecture with SAPT and custom penalty loss function.

A. Pre-processing for Noise Suppression

The PhysioNet CinC Challenge Dataset contains audio recordings of heart rhythm that were collected from various clinical and non-clinical settings, and are inherently noisy. These noise components, especially high-frequency artifacts, can drastically impair the performance of arrhythmia detection models. In this work, a dual-stage pre-processing pipeline for noise suppression has been used. Initially, a Discrete Wavelet Transform is applied to the raw signals to extract multi-scale time-frequency features that capture sustained rhythm patterns. The Discrete Wavelet Transform of a signal *x(t)* is represented by Equation (1).

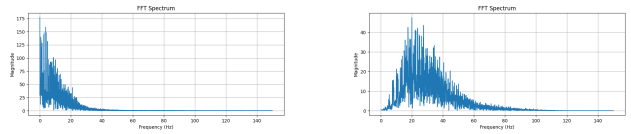
$$W(j,k) = \int_{-\infty}^{\infty} x(t)\,\psi_{j,k}(t)\,dt \tag{1}$$

where $\psi_{j,k}(t)$ are the scaled and translated versions of the original function [13]. The Daubechies 4 (db4) mother waveform was used for this purpose, which is commonly used for signal processing tasks due to its good time-frequency localization properties.

Further, an Infinite Impulse Response (IIR) filter was applied to these transformed signals to smooth the wavelet-processed heart audio and suppress high-frequency noise [14].

To standardize the input size and generalize over variable-length audios, we have segmented each heart audio sample into fixed-length 5-second clips. This specific clip duration was chosen to align with the methodology and positive results reported by [4].

Figure 3 illustrates the incremental effect of the pre-processing pipeline in denoising the raw audio input signal. The final signal retains key rhythmic characteristics and is a cleaner input to the deep learning model. Marked variation is observed in the Fast-Fourier-Transforms of the input and transformed signals as shown in Figure 4



(a) Normal signal (b) Filtered signal

Fig. 4: Comparison of FFTs of the input and filtered signal.

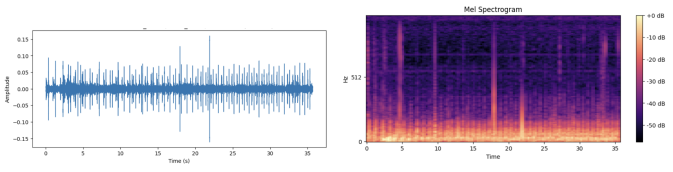


Fig. 5: Conversion to Mel Spectrogram

B. Conversion to Mel Spectrogram

Initial experiments conducted on audio-specific models taking the raw audio files as input did not perform well (discussed further in Section IV). This is attributed to the failure of audio models to recognise the spatial features of the heart rhythm, prompting an exploration of CNN-based image processing models with Mel spectrogram inputs. Mel spectrograms are compact and perceptually relevant descriptions of audio signals. Conversion into Mel spectrogram is ideal for classification tasks because of reduction in dimensionality compared to raw audio waveforms [15]. A Mel spectrogram condenses information into fewer, more meaningful bands instead of working with high-resolution frequency spectrums. Further, Mel spectrograms make the models more robust to variations in background noises in the heart rhythm from the equipment [16].

Figure 5 illustrates the transformation of an audio waveform into its Mel spectrogram representation. The left panel shows the raw audio waveform, displaying amplitude variations over time. The right panel presents the corresponding Mel spectrogram that is provided as input to the model.

C. Proposed Model

The input data used for the model in this study, Mel spectrograms, have evident spatial and temporal properties. The proposed architecture introduces a novel method of exploiting the memory retention capabilities of an LSTM network by modifying the internal gate mechanism of the LSTM block. In particular, the default forget gate of the LSTM is substituted with a parameterized H_∞ filter [17] to develop a new recurrent unit that is referred to as the H_∞ LSTM cell. The H_∞ filter excels in minimizing the worst-case estimation error and is particularly useful when underlying noise is non-Gaussian or of unknown nature [18]. This extension thus allows for better adaptive memory forgetting control for small and noisy datasets in comparison to the traditional forget gate.

The model architecture begins with an input layer designed to receive Mel spectrograms of size (N_{mels}, T, 1) where N_{mels} represents number of Mel filterbanks and T represents time steps. This input is passed through a series of convolutional blocks. Each convolutional block comprises two 3×3 convolutional layers with "same" padding to preserve the dimensionality of the input and followed by a Batch Normalization layer. Batch normalization normalizes across a mini-batch of activations, hence stabilizing and accelerating the training process [19]. Then, at the end of each convolutional block, a 2×2 MaxPooling layer reduces the spatial dimensions, thus effectively summarizing and focusing the network's attention on the most important features.

The resulting feature maps are then reshaped along the time axis and passed into the novel H_∞-LSTM module. In this module, each LSTM cell works similarly to a standard LSTM when computing the input and output gates. However, instead of using the usual forget gate, a learnable filter coefficient λ_h is introduced, which comes from integrating the H_∞ filter into the LSTM's gate design. This coefficient helps the model decide how much weight to give to the previous memory state versus the new input, allowing it to learn a more robust, data-driven forgetting mechanism.

The H_∞ filter is designed to provide a guaranteed bound on estimation error even under unknown-but-bounded disturbances and model inaccuracies, making it well-suited for environments with unpredictable or non-stationary noise. Equations 2-5 describe the input gate (i_t), forget gate (f_t), output gate (o_t), and candidate cell state ($\tilde{c}_t$) respectively for each block of a standard LSTM.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$

where x_t is the input vector at time t, h_{t-1} is the hidden state from time step $t-1$, W_i, W_f, W_o, W_c represent the weights of the input matrices, U_i, U_f, U_o, U_c represent weights of the recurrent matrices, b_i, b_f, b_o, b_c are bias terms, σ is the sigmoid activation function, and $\tanh$ is the hyperbolic tangent activation function [20]. In the proposed architecture, the input and output gate logic is retained, but the forget gate of a standard LSTM is replaced with a mechanism inspired by the H_∞ filtering approach, creating a robustness coefficient λ_h that is derived by passing a trainable parameter K_{filter} through a sigmoid activation:

$$\lambda_h = \sigma(K_{\text{filter}}) \tag{6}$$

This coefficient dynamically controls the trade-off between retaining past memory c_{t-1} and incorporating new information $i_t \tilde{c}_t$, leading to a modified cell state update:

$$c_t = (1 - \lambda_h) c_{t-1} + \lambda_h i_t \tilde{c}_t \tag{7}$$

This structure is mathematically same as the cell state update from the forget gate in a standard LSTM:

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \tag{8}$$

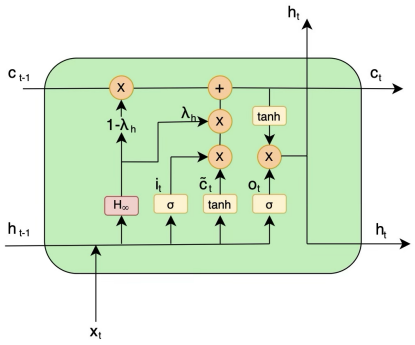


Fig. 6: Structure of a H_∞-LSTM Cell unit

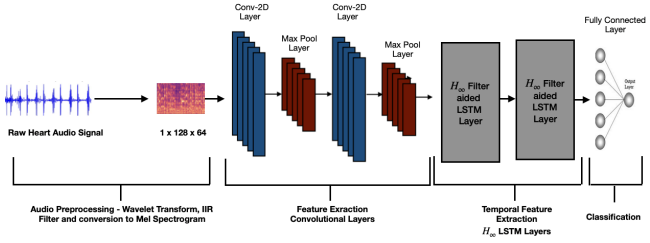


Fig. 7: Proposed Methodology pipeline

except that instead of computing the forget gate f_t as a function of the current input and hidden state, the model directly learns a fixed robustness coefficient λ_h.

Figure 6 illustrates the structure of a H_∞-LSTM Cell unit. In contrast to a regular LSTM unit, the H_∞ filter is used in place of the traditional forget gate and is used to control the cell state update across training. The complete proposed model architecture and training pipeline is demonstrated in Fig 7.

D. Training Methodology

1) Penalty Weighted Loss (PWL): The core idea behind Penalty Weighted Loss (PWL) is to dynamically adjust the contribution of each sample to the loss based on the model's misclassification behavior, particularly focusing on false negatives and false positives, which are critical in medical diagnosis [21]. Given a batch of size B, let the predicted probability vector be $\hat{\mathbf{y}} = (\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^B)$ and the ground truth labels be $\mathbf{y} = (y^1, y^2, \ldots, y^B)$, where $y^i \in \{0, 1\}$. Define a decision threshold $\delta \in [0, 1]$.

a) False Negative Index (FNI)::

$$\text{FNI}(\delta) = \sum_{i=1}^{B} \mathbb{I}(\hat{y}_i \leq \delta \wedge y_i = 1) \tag{9}$$

b) False Positive Index (FPI)::

$$\text{FPI}(\delta) = \sum_{i=1}^{B} \mathbb{I}(\hat{y}_i \geq \delta \wedge y_i = 0) \tag{10}$$

Here, $\mathbb{I}(\cdot)$ denotes the indicator function, that evaluates to 1 when the specified condition holds true, and 0 otherwise.

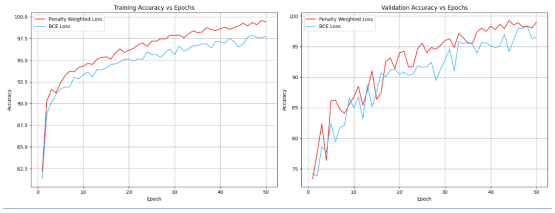


Fig. 8: Comparison of the performance of CNN-H_∞-LSTM model trained under various loss functions

To penalize misclassifications adaptively, the penalty term is defined as:

$$\mathcal{R}_{\text{penalty}}(\delta) = 1 + \alpha \cdot \text{FNI}(\delta) + (1 - \alpha) \cdot \text{FPI}(\delta) \tag{11}$$

where $\alpha \in (0, 1)$ balances the emphasis between false negatives and false positives.

The standard Binary Cross-Entropy (BCE) loss over the batch is:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{B} \sum_{i=1}^{B} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \tag{12}$$

The proposed Penalty Weighted Loss (PWL) becomes:

$$\mathcal{L}_{\text{PWL}} = \mathcal{R}_{\text{penalty}}(\delta) \cdot \mathcal{L}_{\text{BCE}} \tag{13}$$

This increases the loss proportionally to the number of misclassifications, thus compelling the model to focus more on minority class errors, particularly false negatives, which are crucial in clinical settings. Figure 8 compares the training and validation performance of the same proposed model when the custom Penalty Weighted Loss (PWL) is used in comparison to the Binary Cross Entropy Loss described in Equation (14). A marked increase in accuracy for both training and validation is observed using PWL when other training conditions remain the same.

2) Stochastic Adaptive Probe Thresholding (SAPT): While the loss function guides the model to reduce prediction errors, selecting an optimal classification threshold τ is equally vital, especially in imbalanced datasets. A fixed threshold ($\tau = 0.5$) often biases the model towards the majority class, adversely impacting sensitivity metrics such as recall.

SAPT introduces an adaptive framework to dynamically learn the decision threshold τ^* during training over the threshold space $[0, 1]$.

The objective is to find a threshold τ^* that maximizes a task-specific metric $M(\tau)$, such as F1-score, balanced accuracy, or recall.

1) Discretization: At a predefined epoch interval γ, evaluate a set of candidate thresholds $T = \{\tau_j \in [0, 1]\}$
2) Evaluation: At each epoch t, compute the F1-score $F_t(\tau_j)$ for each threshold τ_j on the validation set.
3) Stochastic Optimization: To account for the noise in model predictions, adopt a stochastic approximation approach where thresholds are sampled and updated based on recent performance estimates.

4) Threshold Update: During each γ-epoch window, select the threshold τ_t that maximizes the expected F1-score.
5) Adaptive Decision Boundary: The selected threshold τ_t^* is used in the subsequent epoch for classification, allowing the model to adapt to changes in the data distribution or model calibration during training.

III. Experimental Setup

To ensure fair comparison, a consistent setup has been followed across all models. Each model was trained for 50 epochs with a fixed learning rate of 0.001 using the Adam optimizer. The dataset was partitioned into an 80:20 training and testing split. An additional validation set, comprising 250 healthy and 250 unhealthy samples, was strictly reserved for hyperparameter tuning. To ensure the integrity and generalization of the model, the testing set was entirely isolated from the training process. Further, no data segments originating from the same patient were assigned to the training and testing partitions simultaneously, thereby preventing data leakage. All experiments were conducted using PyTorch version 2.6.0 and Python 3.10, with an Nvidia P100 GPU for hardware acceleration. A batch size of 64 was chosen to maintain a balance between efficiency and stable gradient updates.

A fine-tuning strategy was adopted, where a CNN-LSTM model was trained on the dataset, and then the weights of the CNN layers were transferred to the CNN-H_∞-LSTM model. The CNN layers were then frozen, and the weights of the H_∞-LSTM layers were trained. This isolates the adaptation to the layers that matter most for handling time dependencies under noise and imbalance, while leveraging the pre-learned representations for spatial features.

Further, in the experiments, the Stochastic Adaptive Probe Thresholding (SAPT) hyperparameter γ has been set to 10 based on experimental tuning. This provides the model with sufficient iterations to smooth performance metrics over time as lower values resulted in overly reactive threshold updates, causing erratic behavior in early training epochs. Higher values, on the other hand, slowed the adaptation and reduced its generalization. The Exponential Weighted Moving Average (EWMA) smoothing factor used in this case was 0.3, while the initial classification threshold τ remains set to the default 0.5. Additionally, the hyperparameter α in the Penalty Weighted Loss is set to a value of 0.87, which is roughly equal to the class imbalance ratio of the dataset used.

IV. Results

In this section, a comparative analysis is conducted between the proposed model, audio specific models, vision models and models proposed in heart sound classification literature. Accuracy, F1 score, sensitivity and specificity has been used as metrics to evaluate different models comprehensively.

A. Audio Models

The first experiments were with end-to-end audio-based deep learning models. The Wav2Vec, HuBERT, and DistilHuBERT architectures were tested on the raw audio recordings.

TABLE I: Performance of pre-trained audio models on heart sound classification

Model	F1 Score (%)	Accuracy (%)	Sensitivity (%)	Specificity (%)
Wave2Vec2	69.54	68.14	63.26	63.24
HuBERT	71.21	71.21	71.85	70.58
DistilHuBERT	74.45	78.23	72.47	76.54

Since these models are fine-tuned for speech tasks, they did not generalize well with heart sound data. The classification results were unsatisfactory, and the models showed poor results in distinguishing murmurs. The results are summarized in Table I.

B. Vision Models

The next experiments tested state of the art image classification models with the Log-Mel Spectrograms of the segmented audio clips. Architectures that were considered included ResNet, MobileNet and Vision Transformers. The ResNet-50 model attained the best results with an F1 score of 89.68% and an accuracy of 88.94%. While the Vision Transformer had promising results in specificity, the architecture requires a lot more data to fully leverage its representational power. Table II presents an overview of the obtained results.

C. Proposed Model

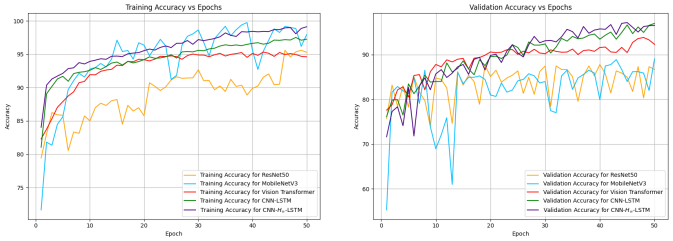
The proposed design augments the CNN-LSTM framework by adding an H_∞ filter in the LSTM cell. The CNN layers identify spatial features, while the LSTM layers track temporal dependencies.
Figs. 9-10 show the improvement in training and validation results achieved by the proposed model and previous studies over successive epochs.

D. Performance comparison with existing approaches

This section compares the proposed model's performance evaluated on the CinC PhysioNet 2016 Dataset. A quantitative

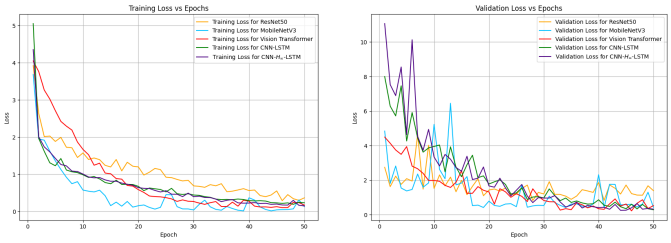
TABLE II: Performance of pre-trained vision models on heart sound classification

Model	F1 Score (%)	Accuracy (%)	Sensitivity (%)	Specificity (%)
ResNet-50	89.68	88.94	93.83	83.82
MobileNetV3-Large	81.36	81.27	79.01	83.82
Vision Transformer	84.23	95.23	89.97	96.08



(a) Training Accuracy (b) Validation Accuracy

Fig. 9: Accuracy Curves For benchmarked models



(a) Training Loss (b) Validation Loss

Fig. 10: Loss Curves for benchmarked models

TABLE III: Performance comparison between the proposed model and top-performing baseline models

Model	F1 (%)	Acc. (%)	Sens. (%)	Spec. (%)
CNN-H_∞-LSTM with SAPT	**98.85**	**99.42**	**99.23**	**99.49**
CNN-LSTM with SAPT	96.19	98.16	94.69	99.29
ResNet-50	89.68	88.94	93.83	83.82
MobileNetV3-Large	81.36	81.27	79.01	83.82
Vision Transformer	84.23	95.23	89.97	96.08
Log-Mel VGGNet [5]	–	–	89.5	89.7
LSTM-CNN [8]	91	86	87	82
Capsule Neural Network [4]	91	90	84.87	–
CRNN [9]	98.34	98.34	98.66	98.01

comparison between test performances of proposed model and existing benchmarks is provided in Table III.

The proposed framework achieves an F1 score of 98.85% and accuracy of 99.42%, outperforming the pretrained models and previous studies. The gains in performance can be attributed to the model's specialized architecture and the new training methodology proposed in this work.

V. Conclusion

In this paper, the CNN-H_∞-LSTM is proposed as a novel deep learning architecture for automated arrhythmia detection from heart sound recordings. The proposed method includes using mel spectrogram of the heart sound recordings as input, replacing the traditional forget gate of a CNN-LSTM with an adaptive H_∞ filter to improve robustness against noise and variability. While the traditional CNN-LSTM relies on a fixed forget gate that is susceptible to noisy and variable signals, the integration of the H_∞ filter allows the model to achieve greater robustness and dynamic state correction for improved arrhythmia detection, even under heavier noise. To address extreme class imbalance, a custom training strategy, SAPT, has been introduced which improves convergence stability and minority class recall. Evaluated on the PhysioNet 2016 CinC Challenge dataset, the proposed method achieves 98.85% F1-score, 99.42% accuracy, 99.23% sensitivity, and 99.49% specificity, outperforming prior approaches. The end-to-end design enables real-time deployment on mobile or edge devices, supporting scalable and low-cost cardiac screening. Future work includes integrating the H_∞ filter to more advanced architectures, coupled with centroid-based thresholding and evaluating AutoBalance optimizer for further improvements in class imbalance handling and clinical applicability.

References

[1] "The top 10 causes of death," *World Health Organization*, Aug 2024.

[2] S. Khurshid, S. H. Choi, L.-C. Weng, E. Y. Wang, L. Trinquart, E. J. Benjamin, P. T. Ellinor, and S. A. Lubitz, "Frequency of cardiac rhythm abnormalities in a half million adults," *Circulation: Arrhythmia and Electrophysiology*, vol. 11, Jul 2018.

[3] T. Nilanon, J. Yao, J. Hao, S. Purushotham, and Y. Liu, "Normal / abnormal heart sound recordings classification using convolutional neural network," in *2016 Computing in Cardiology Conference (CinC)*, pp. 585–588, 2016.

[4] Tsai, Liu, Zheng, and Chen, "Heart murmur classification using a capsule neural network," *MDPI Bioengineering 2023*, 2022.

[5] Li, Chang, and Schuller, "Cnn-based heart sound classification with an imbalance-compensating weighted loss function," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2022.

[6] N. E. Singh-Miller and N. Singh-Miller, "Using spectral acoustic features to identify abnormal heart sounds," in *2016 Computing in Cardiology Conference (CinC)*, pp. 557–560, 2016.

[7] R. Vernekar *et al.*, "A novel approach for classification of normal/abnormal phonocardiogram recordings using temporal signal analysis and machine learning," in *Computing in Cardiology*, 2017.

[8] Chen, Xuan, Gu, Liu, and Chen, "Automatic classification of normal–abnormal heart sounds using convolution neural network and long-short term memory," *MDPI Electronics 2022*, vol. 11, no. 8, 2022.

[9] Deng, Meng, Cao, Wang, Zhang, and Fan, "Heart sound classification based on improved mfcc features and convolutional recurrent neural networks," *Neural Networks*, vol. 130, no. 11, 2020.

[10] Shen, Deng, and Yasmin, "H-infinity filtering for speech enhancement," in *4th International Conference on Spoken Language Processing (ICSLP 1996)*, pp. 873–876, 1996.

[11] Clifford, Liu, Moody, and Springer, "Classification of normal/abnormal heart sound recordings: the physionet/computing in cardiology challenge 2016," in *Computing in Cardiology*, vol. 43, 2016.

[12] Y. Tian, H. Yu, and Y. Zhu, "Jitter error detection and compensation method based on h-infinity filter fusion for remote sensing satellite image," in *2024 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, pp. 1–5, 2024.

[13] N. K. Kit, H. U. Amin, and A. R. Subhani, "Discrete wavelet transform based eeg feature extraction and classification for mental stress using machine learning classifiers," in *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2022.

[14] D. Singh, B. K. Singh, and A. K. Behera, "Comparitive study of different iir filter for denoising lung sound," in *2021 6th International Conference for Convergence in Technology (I2CT)*, pp. 1–3, 2021.

[15] J. Martinsson and M. Sandsten, "Dmel: The differentiable log-mel spectrogram as a trainable layer in neural networks," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5005–5009, 2024.

[16] S. Donkada, S. Pouriyeh, R. M. Parizi, C. Y. Xie, and H. Shahriar, "Early heart disease detection using mel-spectrograms and deep learning," in *2023 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, 2023.

[17] Luan, Liu, and Shi, "H filtering for nonlinear systems via neural networks," in *Journal of the Franklin Institute*, 2009.

[18] T. Moir, "Adaptive crosstalk-resistant noise-cancellation using h infinity filters," in *2019 IEEE International Conference on Signals and Systems (ICSigSys)*, pp. 24–28, 2019.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[20] Hochreiter and Schmidhuber, "Long short-term memory," in *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[21] M. R. Rezaei-Dastjerdehei, A. Mijani, and E. Fatemizadeh, "Addressing imbalance in multi-label classification using weighted cross entropy loss function," in *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*, pp. 333–338, 2020.