A Systematic Evaluation of Hybrid CNN Architectures for Atrial Fibrillation Episode Prediction Using ECG Data

Mbithe Nzomo[1,2][0000−0002−2923−8333] and Deshendran Moodley[1,2][0000−0002−4340−9178]

[1] University of Cape Town, Cape Town, South Africa
[2] Centre for Artificial Intelligence Research (CAIR), Cape Town, South Africa
mnzomo@cs.uct.ac.za and deshen.moodley@uct.ac.za

Abstract. Predicting atrial fibrillation (AF) episodes before onset can enable timely intervention. This study systematically evaluates three hybrid CNN architectures (CNN-BiGRU, CNN-BiLSTM, CNN-Transformer) for AF episode prediction against a CNN baseline. Using the IRIDIA-AF dataset, we compared the architectures across two ECG input window sizes (5 and 10 minutes) and five prediction horizons (5 minutes to 1 hour). While the baseline CNN remained competitive, it was generally outperformed by the hybrid models, particularly at the longer horizons. Although no single hybrid model dominated outright, the CNN-BiLSTM was the most consistently competitive, ranking best in six of the ten configurations. We further investigated the impact of transfer learning on the smaller AFDB dataset, with and without fine-tuning. We found that transfer learning improved AUROC at short to medium horizons (5 to 30 minutes), with the CNN benefitting the most. However, at longer horizons, models trained from scratch on AFDB performed comparably or better. Finally, we analysed the hybrid models using eight experimental configurations from six related AF prediction studies. We outperformed existing studies in six of the eight configurations. Of the three models, the CNN-Transformer performed best in four of the eight configurations.

Keywords: Hybrid CNN · ECG · Atrial fibrillation · Episode prediction.

1 Introduction

Atrial fibrillation (AF) is the most commonly occurring cardiac arrhythmia. 90% of AF patients have symptoms that can be disabling [2]. Predicting AF episodes before they occur is highly beneficial, as AF patients can be alerted in advance to have medication ready to mitigate symptoms, while also relieving the anxiety associated with unanticipated episodes. This is particularly useful for paroxysmal AF, which is characterised by spontaneous, intermittent episodes [2].

Recent advances in deep learning and electrocardiogram (ECG)-capable wearable sensors have enabled systems that can continuously monitor heart rhythm and provide early warnings of AF episodes. Convolutional neural networks (CNNs)

have emerged as the dominant architecture in deep learning-based AF episode prediction using ECG data [1,10,9]. More recently, other architectures have also been explored, including recurrent neural networks (RNNs) such as long short-term memory networks (LSTMs) [6], as well as hybrid architectures combining CNNs with bidirectional gated recurrent units (CNN-BiGRUs) [5,3]. While CNN-BiLSTM and CNN-Transformer architectures have shown promising results in related tasks, such as AF episode detection [11,7], they have not been explored for the AF episode prediction task. Additionally, AF episode prediction datasets are often limited in record number and length, yet the utility of transfer learning to address this has not been assessed for this task. Furthermore, the temporal parameters used in episode prediction, such as input window and prediction horizon, differ between studies, making a fair comparison of different models with existing work challenging.

This study aims to address these gaps through a systematic evaluation of CNN, CNN-BiGRU, CNN-BiLSTM, and CNN-Transformer architectures for ECG-based AF episode prediction. We make three main contributions. Firstly, we propose a standard evaluation pipeline for this task, using two input window sizes and five prediction horizons. We also replicate the temporal parameters used in existing studies, resulting in eight configurations with varying input windows and prediction horizons. Secondly, we report on the performance of the CNN-BiLSTM and CNN-Transformer, which have not been used in previous studies for this task, in comparison to the CNN-BiGRU and CNN. We use IRIDIA-AF [4], a publicly available expert-annotated ECG dataset with 24-hour records. Finally, we investigate the utility of transfer learning by pre-training the models on IRIDIA-AF and testing them on the smaller MIT-BIH Atrial Fibrillation Database (AFDB) [8]. We compare transfer learning with and without fine-tuning against models trained from scratch on the AFDB dataset. All code is publicly available on GitHub.[3]

2 Background and Related Work

AF episode prediction identifies precursor patterns in an ECG segment that indicate whether AF will occur in some future segment. This is distinct from, and more challenging than, episode detection, whereby the ECG segment itself is classified as either containing AF or not. Recent work on AF episode prediction has taken two distinct approaches. The first is classification, which classifies the input based on whether AF will occur in a future segment of the same ECG recording. The prediction horizon, i.e. how far ahead the prediction is made, can be specified in advance, allowing for an explicit and predictable warning time. The second approach is risk estimation, whereby the model outputs a time-varying risk score that is monitored against a threshold. A score exceeding the threshold triggers a warning, thus it becomes a binary decision, with the prediction horizon determined retrospectively based on when the warning was

[3] `https://github.com/mbithenzomo/hybrid_cnn_architectures`

triggered. In the remainder of this section, we review existing studies on deep learning for AF episode prediction, including both approaches.

Majority of existing studies have followed the classification approach, with CNNs and their variations being the dominant architectures. Gilon et al. [3] used a CNN-BiGRU on the IRIDIA-AF dataset to predict episodes using ECG RR intervals, exploring different combinations of input window and prediction horizon. Grégoire et al. [5] extended this work on the same dataset by evaluating longer prediction horizons. Rooney et al. [9] applied a CNN on ECG data from PhysioNet Long Term AF Database (LTAFDB). A CNN was also used by Tzou et al. [10] to analyse P-wave segments from two types of recordings from a private dataset: ECG and neuECG, a relatively recent method that simultaneously records ECG and peripheral sympathetic nerve activity.

Two recent studies took the risk estimation approach. The first is Li et al. [6], who built an early warning pipeline for paroxysmal AF episodes. They pre-trained on the Chapman-Shaoxing, PTB-XL, and Georgia ECG Challenge databases, then trained on the IRIDIA-AF, PhysioNet Paroxysmal AF Prediction, and PhysioNet Normal Sinus Rhythm databases. They began by using a BiLSTM as a feature extractor to classify an ECG segment into one of three states: normal, precursor, and abnormal. They then represented risk as a weighted sum of the precursor and abnormal states. Similarly, Gavidia et al. [1] applied a CNN to classify an ECG into three states: sinus rhythm, pre-AF, and AF, using a privately collected dataset. From these states, the "probability of danger" of an imminent AF episode was computed. None of these studies used a CNN-Transformer or CNN-BiLSTM, although one used a BiLSTM.

3 Methodology

3.1 Problem Definition

We define AF episode prediction as a binary ECG classification problem. Let $\mathbf{X} = \{x_1, x_2, x_3, \ldots, x_T\}$ be an ECG signal where x_t is the observation at time t and T is the total length of the signal. We can segment the time series into input window ($\mathbf{W}^{\text{input}}$), the historical ECG data used for prediction, and target window ($\mathbf{W}^{\text{target}}$), the segment of future ECG data where AF episodes may occur. Formally, $\mathbf{W}^{\text{input}} = \{x_{t-w_1}, x_{t-w_1+1}, \ldots, x_t\}$, where w_1 is the input window size, and $\mathbf{W}^{\text{target}} = \{x_{t+h+1}, x_{t+h+2}, \ldots, x_{t+h+w_2-1}, x_{t+h+w_2}\}$, where h is the prediction horizon, and w_2 is the target window size. The classification label y is either 1 if AF occurs in $\mathbf{W}^{\text{target}}$ or 0 otherwise. Thus, the goal of AF episode prediction is to learn a function f that maps from $\mathbf{W}^{\text{input}}$ to y.

3.2 Data Preprocessing

The IRIDIA-AF dataset contains 167 ECG records from 152 patients, all of whom have paroxysmal AF. It has been used in several prior studies for this task, is expert-annotated, and contains 24-hour records, significantly longer than other

available datasets. For the transfer learning experiments, we selected AFDB as the target dataset. Containing only 23 complete recordings of 10 hours each, it is well-suited to benefit from transfer learning.

To preprocess the data, we began by applying a bandpass Butterworth filter to remove noise. As illustrated in Fig. 1, we then used a sliding window approach to extract samples from the ECG data, aligning with several existing studies on AF episode prediction [3,5,9]. This involves specifying the input window size, target window size, prediction horizon, and step size, which determines the overlap between consecutive windows. We used five prediction horizons ranging from five minutes to one hour, keeping the other parameters fixed.

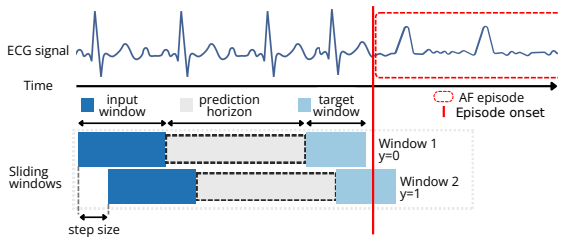

Fig. 1: Sliding windows segmenting ECG data for AF episode prediction.

3.3 Model Architectures

The baseline CNN architecture was adapted from the example given in the IRIDIA-AF dataset documentation.[4] This original architecture consists of three groups of three residual blocks. The number of output channels (c) and kernel sizes (k) are fixed: Group 1 ($c = 16$, $k = 7$), Group 2 ($c = 32$, $k = 5$), and Group 3 ($c = 64$, $k = 3$). Each residual block consists of two one-dimensional convolutional layers, each followed by ReLU activation and batch normalisation.

Although we retained this general structure, we made several key modifications. First, we made the learning rate, weight decay, output channels, and kernel sizes configurable, enabling the hyperparameter optimisation in Section 3.4. Second, we adopted a less aggressive downsampling strategy to preserve temporal resolution. Instead of 2× downsampling per residual block (512× overall) as in the original design, we apply 4× max pooling with dropout after Groups 1 and 2, yielding only a 16× reduction. This retains more fine-grained temporal information in the earlier layers. Third, we applied adaptive average pooling with dropout after Group 3, collapsing the sequence to a fixed length and making the architecture agnostic to input size. Finally, the architecture returns raw logits rather than probabilities, enabling *BCEWithLogitsLoss*, which applies the sigmoid internally for improved numerical stability during training. Probabilities were obtained later at inference by applying the sigmoid function to the output.

[4] `https://github.com/cedricgilon/iridia-af/blob/main/examples/dl/model.py`

The CNN-BiLSTM, CNN-BiGRU, and CNN-Transformer build on the CNN architecture. After the adaptive average pooling stage, the resulting feature map is transposed. For the CNN-BiLSTM and CNN-BiGRU variants, it is then processed by a bidirectional LSTM or GRU respectively, allowing the model to capture temporal dependencies in both directions. For the CNN-Transformer variant, a learned positional embedding is first added to the sequence before it is processed by a Transformer encoder. We then apply mean and max pooling over the full output and concatenate the two pooled representations, which are then fed through the two fully connected layers. Fig. 2 illustrates the architecture of the hybrid CNN models, which share the same CNN backbone.

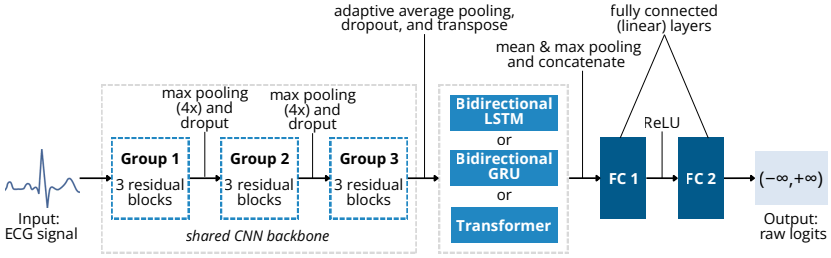

Fig. 2: Hybrid CNN architecture.

3.4 Hyperparameter Optimisation, Training, and Evaluation

Hyperparameter optimisation was performed for the baseline CNN using Optuna,[5] a Bayesian optimisation framework, with the objective of minimising the Brier score on the validation set. The Brier score is a proper scoring rule that evaluates probabilistic predictions by calculating the mean squared error between predicted probabilities and true values, ensuring the model's estimated probabilities are consistent with the true outcome. As shown in Table 1, five CNN hyperparameters were tuned for each dataset: dropout rate, learning rate, weight decay, adaptive pool size, base output channels and kernel size.

Table 1: CNN hyperparameter search spaces and selected values for each prediction horizon H.

Dataset (input)	H	Dropout [0.2–0.5]	Learn. rate [1e-5–1e-2]	Wt. decay [1e-6–1e-3]	Pool size [512/1024/2048]	Channels [16/32/64]	Kernel [5/7]
IRIDIA-AF	5	0.3	1.7e-3	2.2e-5	512	32	7
(5 min)	15	0.2	4.7e-3	2.5e-6	512	16	5
	30	0.2	2.3e-3	3.1e-4	2048	16	5
	45	0.2	1.3e-3	9.0e-5	1024	32	7

Continued on next page.

[5] https://optuna.org/

Dataset (input)	H	Dropout [0.2–0.5]	Learn. rate [1e-5–1e-2]	Wt. decay [1e-6–1e-3]	Pool size [512/1024/2048]	Channels [16/32/64]	Kernel [5/7]
	60	0.2	3.8e-3	1.5e-5	512	64	5
IRIDIA-AF (10 min)	5	0.3	3.76e-3	9.34e-5	1024	16	7
	15	0.2	3.27e-3	1.28e-4	1024	16	7
	30	0.4	3.26e-3	1.19e-5	512	32	7
	45	0.2	6.62e-3	2.25e-4	512	32	7
	60	0.2	2.00e-3	2.23e-4	512	32	7
AFDB (5 min)	5	0.2	2.7e-4	1.2e-4	512	32	7
	15	0.2	7.2e-4	5.0e-6	512	32	5
	30	0.2	2.4e-4	2.2e-5	512	32	7
	45	0.2	2.8e-3	1.7e-5	512	16	5
	60	0.4	7.0e-3	1.0e-6	512	32	7

We trained each model using 5-fold cross-validation with patient-level stratification to prevent data leakage. We used a 64:16:20 ratio for training, validation, and testing per fold. Weighted random sampling was applied during training to address class imbalance; the validation and test sets were left imbalanced. Each fold used early stopping with a patience of 5 epochs, monitoring validation loss to prevent overfitting. To stabilise training, we used a linear learning rate warmup over the first 3 epochs and applied gradient clipping to prevent exploding gradients. Post-hoc probability calibration was done using isotonic regression to ensure the predicted probabilities reliably reflect observed likelihood of upcoming episodes.

The classification threshold was selected on the validation set after training, leaving the test set only for the final evaluation. To select the threshold, we prioritised recall by optimising for the F2 score (i.e. weighting recall 2× more than precision), while constraining the difference between recall and precision to no more than 0.25 to ensure a reasonable precision. If no threshold satisfied this constraint, we fell back to optimising for the F1.5 score. We report five metrics to evaluate model performance: area under the receiver operating characteristic curve (AUROC), and area under the precision-recall curve (AUPRC), recall, F1 score, and specificity. AUROC and AUPRC summarise model performance across all possible classification thresholds, with AUROC measuring ranking ability and AUPRC measuring the trade-off between precision and recall. Recall captures the proportion of AF episodes that were correctly predicted, reflecting the clinical priority of reducing missed episodes. On the other hand, precision measures the proportion of predicted AF episodes that were actually correct. The F1 score is the harmonic mean of recall and precision. Finally, specificity measures the proportion of non-AF episodes that were correctly predicted.

4 Experiments and Results

4.1 Comparison of Input Window Size and Prediction Horizon

We first compared the performance of the four models across two input windows (5 and 10 minutes) and five prediction horizons (5, 15, 30, 45, and 60 minutes) using the IRIDIA-AF dataset. We used a 30-second target window to align with

the clinical definition of the minimum duration of an AF episode [2]. We selected a 70% overlap (1.5-minute step size for the 5-minute input and 3-minute step size for the 10-minute input), balancing temporal coverage with computational efficiency. The results for the 5- and 10-minute input windows are presented in Tables 2 and 3 respectively, which show the per-metric mean and 95% confidence interval computed across five folds. The best performing model was selected based on AUROC, AUPRC, recall, F1 score, and specificity, in that order. This prioritises threshold-agnostic measures of performance (AUROC and AUPRC), while acknowledging the clinical importance of reducing missed episodes (recall).

Increasing the input window from 5 to 10 minutes yielded inconsistent improvement across horizons and models. At short to medium horizons, the two input windows were effectively equivalent for all models. The biggest gains were observed at the 45-minute horizon for the CNN-BiGRU and CNN-BiLSTM, and at the 60-minute horizon for the CNN-Transformer, where every metric improved with the longer input window, with the exception of specificity for the CNN-BiGRU. These improvements were not mirrored by the CNN, which remained largely insensitive to input window size across all horizons.

Unsurprisingly, and consistent with prior studies [3,9], performance generally decreased with increasing prediction horizon across the board. The baseline CNN was never the outright best model, but remained competitive overall. However, it generally fell behind the other architectures as the horizon lengthened. No single hybrid model dominated outright. The CNN-BiGRU and CNN-BiLSTM led at short to medium horizons (5 to 30 minutes), while the CNN-Transformer only overtook them at the longest horizon and input window. However, the CNN-BiLSTM was the most consistently competitive architecture, ranking best in six of the ten dataset configurations.

Table 2: Results for 5-minute input window (IRIDIA-AF dataset) with 95% CI. Best mean score per prediction horizon (H) is **highlighted**, and best model is underlined.

H	Model	AUROC	AUPRC	Recall	F1 Score	Specificity
5	CNN	0.97 [0.96,0.98]	**0.92** [0.88,0.94]	0.94 [0.90,0.97]	0.90 [0.88,0.92]	0.95 [0.93,0.97]
	CNN-BiGRU	**0.98** [0.97,0.98]	0.91 [0.88,0.94]	0.95 [0.94,0.97]	**0.91** [0.89,0.93]	**0.96** [0.94,0.97]
	CNN-BiLSTM	0.97 [0.95,0.98]	**0.92** [0.88,0.95]	**0.96** [0.95,0.97]	**0.91** [0.89,0.93]	**0.96** [0.94,0.97]
	CNN-Transf.	0.97 [0.96,0.98]	0.90 [0.86,0.93]	0.93 [0.91,0.94]	0.89 [0.87,0.91]	**0.96** [0.94,0.97]
15	CNN	**0.96** [0.95,0.96]	0.87 [0.85,0.89]	0.91 [0.89,0.94]	0.86 [0.83,0.88]	0.94 [0.92,0.96]
	CNN-BiGRU	**0.96** [0.93,0.97]	0.87 [0.83,0.90]	**0.93** [0.92,0.95]	**0.88** [0.86,0.91]	**0.95** [0.93,0.97]
	CNN-BiLSTM	**0.96** [0.93,0.97]	0.87 [0.84,0.90]	**0.93** [0.90,0.95]	**0.88** [0.87,0.90]	**0.95** [0.93,0.96]
	CNN-Transf.	0.95 [0.94,0.96]	**0.88** [0.86,0.90]	0.91 [0.90,0.93]	0.86 [0.86,0.87]	0.93 [0.90,0.95]
30	CNN	**0.93** [0.92,0.93]	0.80 [0.78,0.83]	0.86 [0.83,0.88]	0.79 [0.77,0.82]	0.91 [0.88,0.94]
	CNN-BiGRU	0.92 [0.90,0.93]	0.79 [0.76,0.80]	0.83 [0.82,0.86]	0.80 [0.78,0.82]	**0.93** [0.88,0.95]
	CNN-BiLSTM	**0.93** [0.92,0.94]	**0.81** [0.78,0.84]	**0.88** [0.83,0.89]	**0.82** [0.76,0.84]	0.92 [0.87,0.95]
	CNN-Transf.	0.92 [0.91,0.93]	0.79 [0.75,0.82]	0.86 [0.84,0.88]	0.76 [0.70,0.81]	0.90 [0.89,0.92]
45	CNN	**0.90** [0.89,0.90]	0.72 [0.71,0.74]	**0.82** [0.80,0.83]	**0.73** [0.69,0.77]	**0.88** [0.85,0.90]
	CNN-BiGRU	0.89 [0.87,0.90]	0.72 [0.65,0.75]	0.78 [0.74,0.80]	0.71 [0.68,0.75]	**0.88** [0.84,0.91]
	CNN-BiLSTM	0.88 [0.86,0.89]	0.70 [0.67,0.72]	0.79 [0.76,0.82]	0.71 [0.69,0.75]	0.87 [0.85,0.89]
	CNN-Transf.	**0.90** [0.89,0.91]	**0.74** [0.71,0.78]	0.81 [0.81,0.82]	**0.73** [0.68,0.76]	0.85 [0.82,0.90]
60	CNN	0.85 [0.84,0.87]	0.65 [0.61,0.69]	0.75 [0.74,0.77]	0.67 [0.61,0.71]	0.85 [0.82,0.89]
	CNN-BiGRU	0.86 [0.85,0.87]	**0.68** [0.66,0.70]	0.76 [0.72,0.79]	0.70 [0.67,0.72]	0.87 [0.84,0.91]

Continued on next page.

H	Model	AUROC	AUPRC	Recall	F1 Score	Specificity
	CNN-BiLSTM	**0.88** [0.87,0.89]	**0.68** [0.65,0.73]	**0.80** [0.78,0.86]	**0.74** [0.71,0.77]	**0.90** [0.88,0.92]
	CNN-Transf.	0.86 [0.84,0.87]	0.65 [0.60,0.70]	0.78 [0.73,0.83]	0.66 [0.60,0.71]	0.82 [0.76,0.88]

Table 3: Results for 10-minute input window (IRIDIA-AF dataset) with 95% CI. Best mean score per prediction horizon (H) is **highlighted**, and best model is underlined.

H	Model	AUROC	AUPRC	Recall	F1 Score	Specificity
5	CNN	0.96 [0.95,0.98]	0.91 [0.88,0.93]	0.94 [0.92,0.97]	0.89 [0.87,0.92]	0.95 [0.93,0.97]
	CNN-BiGRU	0.97 [0.96,0.98]	0.91 [0.88,0.95]	**0.95** [0.93,0.96]	**0.91** [0.89,0.93]	**0.96** [0.95,0.98]
	CNN-BiLSTM	**0.98** [0.96,0.99]	**0.94** [0.89,0.96]	**0.95** [0.93,0.97]	**0.91** [0.89,0.93]	**0.96** [0.94,0.98]
	CNN-Transf.	0.97 [0.95,0.98]	0.90 [0.85,0.94]	0.93 [0.92,0.95]	0.88 [0.83,0.91]	0.95 [0.93,0.97]
15	CNN	0.95 [0.93,0.96]	0.85 [0.82,0.88]	0.91 [0.90,0.92]	0.86 [0.84,0.88]	**0.94** [0.92,0.96]
	CNN-BiGRU	**0.96** [0.95,0.97]	0.87 [0.81,0.90]	0.90 [0.89,0.92]	0.85 [0.82,0.88]	**0.94** [0.91,0.96]
	CNN-BiLSTM	**0.96** [0.93,0.97]	**0.88** [0.84,0.90]	0.92 [0.90,0.95]	**0.88** [0.85,0.89]	**0.94** [0.92,0.97]
	CNN-Transf.	0.95 [0.93,0.96]	0.86 [0.82,0.89]	**0.93** [0.91,0.94]	0.84 [0.82,0.86]	0.92 [0.90,0.93]
30	CNN	0.92 [0.90,0.93]	0.79 [0.75,0.82]	**0.87** [0.84,0.90]	0.78 [0.68,0.82]	0.89 [0.84,0.93]
	CNN-BiGRU	0.90 [0.87,0.93]	0.75 [0.71,0.82]	0.83 [0.74,0.86]	**0.79** [0.75,0.84]	**0.92** [0.89,0.95]
	CNN-BiLSTM	**0.93** [0.90,0.94]	**0.80** [0.74,0.83]	**0.87** [0.83,0.91]	0.78 [0.72,0.82]	0.89 [0.83,0.93]
	CNN-Transf.	0.92 [0.90,0.93]	0.78 [0.75,0.80]	0.83 [0.79,0.87]	0.78 [0.74,0.80]	0.91 [0.90,0.93]
45	CNN	0.89 [0.87,0.90]	0.74 [0.70,0.76]	0.82 [0.79,0.85]	0.73 [0.69,0.78]	0.88 [0.85,0.92]
	CNN-BiGRU	**0.91** [0.89,0.92]	**0.77** [0.73,0.80]	**0.85** [0.83,0.87]	0.76 [0.73,0.79]	0.88 [0.85,0.91]
	CNN-BiLSTM	0.90 [0.88,0.91]	0.75 [0.72,0.77]	0.80 [0.75,0.85]	**0.77** [0.76,0.78]	**0.92** [0.89,0.94]
	CNN-Transf.	0.90 [0.89,0.92]	0.76 [0.71,0.80]	0.82 [0.79,0.85]	0.75 [0.72,0.79]	0.88 [0.84,0.92]
60	CNN	0.86 [0.83,0.87]	0.66 [0.61,0.69]	0.76 [0.72,0.77]	**0.68** [0.58,0.72]	0.86 [0.81,0.88]
	CNN-BiGRU	0.86 [0.84,0.88]	0.66 [0.62,0.70]	0.78 [0.72,0.82]	0.66 [0.64,0.72]	0.84 [0.82,0.86]
	CNN-BiLSTM	0.88 [0.86,0.89]	**0.70** [0.66,0.76]	**0.79** [0.76,0.83]	**0.68** [0.63,0.73]	0.84 [0.79,0.89]
	CNN-Transf.	**0.89** [0.88,0.89]	**0.71** [0.69,0.72]	**0.79** [0.73,0.83]	**0.72** [0.71,0.73]	**0.88** [0.85,0.91]

4.2 Transfer Learning

Using a 5-minute input window, we assessed the impact of transfer learning on the models' overall ranking ability across thresholds. To establish the baseline, the models were trained from scratch and tested on the AFDB dataset, following the same methodology described in Section 3.4. We then re-trained the four models on the full IRIDIA-AF dataset. We tested the performance of the models when directly transferred to the AFDB dataset with no fine-tuning. Then, we tested the performance of the models fine-tuned on the AFDB dataset. Fine-tuning used a two-stage approach: first, the CNN backbone was frozen and only the head was trained at the full learning rate for some epochs; and then the entire network was unfrozen and trained at a reduced learning rate for the remaining epochs. Table 4 compares the AUROC scores of the models trained solely on AFDB (no transfer) against those trained on the source datasets, with and without fine-tuning on AFDB. The table also shows the performance gain from fine-tuning compared to both no transfer and no fine-tuning.

Compared to the IRIDIA-AF results, the baseline models demonstrate generally poorer performance and model instability as indicated by the large CI ranges. With only 23 patients, 10-hour recordings, and a mix of persistent and paroxysmal AF, the dataset presents additional learning challenges compared

to IRIDIA-AF which has 152 patients, 24-hour recordings, and only paroxysmal AF. Transfer learning improved AUROC across the board, with the exception of the 45- and 60-minute prediction horizons. The improvement delta generally decreased with increasing horizon, even with fine-tuning applied. This indicates that the features learned from IRIDIA-AF transfer well to the AFDB dataset at short to medium horizons, whereas at long horizons, AFDB's own from-scratch training captures the relevant features better. The CNN benefitted the most from transfer learning. It showed the largest performance delta between fine-tuning and no transfer at every horizon except 5 minutes, and between fine-tuning and not fine-tuning at every horizon except 45 minutes.

Table 4: Impact of transfer learning on AUROC for 5-minute input (AFDB dataset). Best mean score per prediction horizon is **highlighted**, and corresponding model is underlined.

Horizon	Model	No Transfer	Not Fine-tuned	Fine-tuned	Δ_1	Δ_2
5	CNN	0.83 [0.77,0.85]	0.74 [0.55,0.91]	0.87 [0.82,0.94]	+0.04	+0.13
	CNN-BiGRU	0.80 [0.75,0.82]	0.81 [0.71,0.90]	0.92 [0.89,0.96]	+0.12	+0.11
	CNN-BiLSTM	0.69 [0.51,0.83]	0.78 [0.68,0.88]	0.84 [0.71,0.94]	+0.15	+0.06
	CNN-Transformer	0.73 [0.68,0.82]	0.85 [0.75,0.94]	**0.93** [0.90,0.96]	+0.20	+0.08
15	CNN	0.71 [0.51,0.80]	0.74 [0.63,0.88]	**0.89** [0.85,0.93]	+0.18	+0.15
	CNN-BiGRU	0.74 [0.62,0.82]	0.68 [0.60,0.76]	0.76 [0.68,0.84]	+0.02	+0.08
	CNN-BiLSTM	0.78 [0.71,0.84]	0.77 [0.64,0.91]	0.75 [0.59,0.86]	-0.03	-0.02
	CNN-Transformer	0.74 [0.65,0.86]	0.76 [0.68,0.88]	0.83 [0.79,0.92]	+0.09	+0.07
30	CNN	0.55 [0.47,0.63]	0.59 [0.51,0.73]	0.74 [0.59,0.84]	+0.19	+0.15
	CNN-BiGRU	0.54 [0.48,0.57]	**0.76** [0.64,0.88]	0.72 [0.61,0.82]	+0.18	-0.04
	CNN-BiLSTM	0.59 [0.43,0.77]	0.69 [0.52,0.77]	0.64 [0.57,0.74]	+0.05	-0.05
	CNN-Transformer	0.55 [0.49,0.64]	0.61 [0.48,0.81]	0.63 [0.54,0.80]	+0.08	+0.02
45	CNN	0.58 [0.51,0.66]	**0.67** [0.52,0.77]	0.62 [0.54,0.71]	+0.04	-0.05
	CNN-BiGRU	0.52 [0.45,0.57]	0.57 [0.43,0.71]	0.52 [0.42,0.68]	+0.00	-0.05
	CNN-BiLSTM	0.62 [0.52,0.73]	0.54 [0.35,0.71]	0.55 [0.44,0.66]	-0.07	+0.01
	CNN-Transformer	0.58 [0.45,0.73]	0.61 [0.55,0.71]	0.59 [0.49,0.70]	+0.01	-0.02
60	CNN	0.66 [0.52,0.77]	0.50 [0.37,0.63]	0.64 [0.51,0.80]	-0.02	+0.14
	CNN-BiGRU	0.65 [0.52,0.79]	0.66 [0.57,0.72]	0.63 [0.53,0.73]	-0.02	-0.03
	CNN-BiLSTM	0.66 [0.51,0.77]	0.62 [0.47,0.72]	0.61 [0.52,0.74]	-0.05	-0.01
	CNN-Transformer	**0.68** [0.62,0.81]	0.48 [0.35,0.64]	0.48 [0.39,0.56]	-0.20	+0.00

Δ_1 = Fine-tuned − No Transfer; Δ_2 = Fine-tuned − Not Fine-tuned.

4.3 Comparison with Existing Studies

We compared the performance of the hybrid CNN models against models from the AF episode prediction studies reviewed in Section 2. We used the IRIDIA-AF dataset to replicate the temporal parameters from these studies. Where the study specified parameters in RR intervals, we assumed a heart rate of 60 beats per minute. If target window or step size was not explicitly stated, we used a default target window of 30 seconds and manually calculated the step size to ensure an overlap of 70%. For the risk estimation studies, we used the mean early warning time as the prediction horizon.

We trained and evaluated the three hybrid models on these dataset configurations, following the same cross-validation methodology detailed in Section 3.4. Table 5 shows the comparison of results. As only one existing study reports

AUPRC, we exclude it from the table. Where an existing study evaluated multiple models, we include only its best-performing model in the table. For each comparison, we report results from whichever of the three hybrid models achieved the highest mean score, prioritising AUROC, then recall, F1 score, and specificity.

The three models achieved generally comparable performance, with scores within a spread of 0.05 across most dataset configurations and metrics. A notable exception was the dataset with a 60-minute horizon and 3-minute input, in which the CNN-Transformer outperformed the other two by a larger margin (AUROC spread: 0.08; specificity spread: 0.13). The CNN-Transformer was the best performing model in four of the eight dataset configurations, the CNN-BiGRU in three configurations, and CNN-BiLSTM in the remaining one.

Our models surpassed the results reported from existing studies in six of the eight experiments. The two exceptions are the datasets replicated from Gavidia et al. [1] and Li et al. [6], in which none of the four models surpassed the reported results. Notably, these studies both took a risk estimation approach to episode prediction. For the 18.9-minute horizon and 10-second input, the BiLSTM by Li et al. [6] outperformed our best model, the CNN-Transformer, in recall (0.86 vs. 0.76), F1 score (0.75 vs. 0.67), and specificity (0.92 vs. 0.85). Although we were able to achieve a similar recall by adjusting the threshold, this came at the cost of significantly lowered specificity; thus we were unable to surpass their reported results. We note that they do not report AUROC or AUPRC, and so we cannot assess the performance of their BiLSTM across thresholds.

For the 32.5-minute horizon and 30-second input, the CNN by Gavidia et al. [1] achieved a higher recall than our best performing model, the CNN-BiLSTM (0.95 vs. 0.89). However, this came at the cost of a relatively low specificity of 0.69, suggesting a threshold choice that prioritised correctly identifying upcoming episodes over avoiding false alarms. At the same time, they reported an AUROC of 0.95 and AUPRC of 0.96, suggesting good discrimination regardless of threshold. According to their supplementary file, the ECG data was resampled to a balanced dataset. Although training on balanced data is methodologically sound, testing on balanced data can inflate performance metrics as this distribution does not match real-world AF episode frequency. We replicated their approach by using weighted random sampling to achieve a balanced test set. We observed an improvement in AUPRC (0.73 to 0.87), recall (0.78 to 0.89), and F1 score (0.72 to 0.81), but this came at the cost of specificity, and we were unable to surpass any of their reported metrics. We include these results in Table 5 alongside those from the imbalanced test set, indicating each clearly.

Table 5: Comparison of the hybrid models against models from existing studies. The best mean scores are in **bold** and the best model is underlined.

Dataset	Model	AUROC	Recall	F1 Score	Specificity
H: 0.5 m	CNN-BiGRU [5]	0.74 [0.73,0.76]	0.80 [0.79,0.82]	0.71 [0.70,0.71]	0.53 [0.51,0.55]
I: 5 m	CNN-BiGRU [3]	0.71 [0.70,0.73]	–	–	–

Continued on next page.

Dataset	Model	AUROC	Recall	F1 Score	Specificity
	Our CNN-BiGRU	**0.98** [0.97,0.99]	**0.96** [0.94,0.98]	**0.91** [0.89,0.94]	**0.96** [0.93,0.98]
H: 5 m I: 5 m	CNN [10]	0.94	0.88	0.88	0.89
	CNN-BiGRU [3]	0.70 [0.69,0.72]	–	–	–
	Our CNN-BiGRU	**0.98** [0.97,0.98]	**0.95** [0.94,0.97]	**0.91** [0.89,0.93]	**0.96** [0.94,0.97]
H: 7.5 m I: 3 m	CNN [9]	0.73 [0.71,0.75]	0.60 [0.57,0.63]	–	–
	Our CNN-BiGRU	**0.96** [0.94,0.97]	**0.94** [0.92,0.96]	0.86 [0.83,0.88]	0.92 [0.90,0.95]
H: 15 m I: 3 m	CNN [9]	0.72 [0.70,0.74]	0.58 [0.56,0.60]	–	–
	Our CNN-Transf.	**0.94** [0.93,0.95]	**0.88** [0.85,0.91]	0.80 [0.75,0.85]	0.90 [0.87,0.94]
H: 18.9 m I: 10 s	BiLSTM [6]	–	**0.86**	**0.75**	**0.92**
	Our CNN-Transf.	0.87 [0.85,0.90]	0.76 [0.69,0.83]	0.67 [0.57,0.77]	0.85 [0.74,0.95]
H: 30 m I: 3 m	CNN [9]	0.72 [0.70,0.74]	0.58 [0.56,0.60]	–	–
	Our CNN-Transf.	**0.90** [0.89,0.90]	**0.84** [0.82,0.87]	0.71 [0.67,0.73]	0.84 [0.82,0.87]
H: 32.5 m I: 30 s	CNN* [1]	**0.95**	**0.95**	–	0.69
	Our CNN-BiLSTM*	0.88 [0.86,0.90]	0.89 [0.86,0.91]	0.81 [0.78,0.83]	0.69 [0.58,0.76]
	Our CNN-BiLSTM†	0.89 [0.88,0.91]	0.78 [0.74,0.86]	0.72 [0.67,0.76]	**0.89** [0.84,0.92]
H: 60 m I: 3 m	CNN [9]	0.71 [0.69,0.73]	0.57 [0.54,0.60]	–	–
	Our CNN-Transf.	**0.76** [0.73,0.78]	**0.66** [0.56,0.74]	0.52 [0.50,0.54]	0.74 [0.65,0.81]

H: Prediction horizon; I: Input window; – Not reported; * Balanced test set; † Imbalanced test set

5 Conclusion

This study presents a systematic comparison of CNN, CNN-BiGRU, CNN-BiLSTM, and CNN-Transformer architectures for ECG-based AF episode prediction. We evaluated model performance across two input window sizes (5 and 10 minutes) and five prediction horizons (5, 15, 30, 45, and 60 minutes). Increasing the input window had little impact at short to medium horizons, suggesting a 5-minute window is sufficient for predicting AF episodes 30 minutes or less in advance. Gains from the increased longer window were concentrated at the two longest horizons, particularly for the hybrid models. All four architectures performed comparably at shorter horizons (5 and 15 minutes), while at longer horizons performance became more architecture-dependent. Overall, the CNN-BiLSTM achieved the most consistently superior performance, followed by the CNN-BiGRU and then the CNN-Transformer, which showed the clearest advantage at the 60-minute horizon with the 10-minute input.

We further investigated the impact of transfer learning from IRIDIA-AF to the smaller AFDB, evaluated based on AUROC score. Pre-training on IRIDIA-AF and fine-tuning on AFDB improved AUROC at short to medium horizons (5–30 minutes). However, this benefit diminished at the two longest horizons, where models trained from scratch on AFDB performed comparably or better. The CNN showed the most consistent gains from transfer learning across all horizons, while the hybrid models' gains were more concentrated at shorter horizons. Finally, we replicated the temporal parameters from six existing AF episode prediction studies, resulting in eight configurations. Our models significantly exceeded previously reported results in six of the eight configurations. Of the three hybrid models, the CNN-Transformer was the best performing model in four configurations, the CNN-BiGRU in three, and CNN-BiLSTM in the remaining one.

This study has some limitations. We focused on comparing hybrid CNN architectures incorporating RNNs and transformers. However, other architectures, including temporal convolutional networks and graph neural networks, have also recently shown promise for AF analysis. Evaluating these architectures for AF episode prediction remains an open question for future work. Future work will also explore alternative configurations of the recurrent and attention-based modules, for example, varying the number of layers, hidden dimensions, or attention heads, which were held fixed in this study to isolate the effect of architecture type. Additionally, this study does not provide explanations for the model outputs. Future research will focus on explainability, which will further enhance the decision support provided by these models.

Acknowledgments. Computations for this study were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: `https://ucthpc.uct.ac.za`.

References

1. Gavidia, M., et al.: Early warning of atrial fibrillation using deep learning. Patterns **5**(6), 100970 (2024)
2. van Gelder, I.C., et al.: 2024 ESC Guidelines for the management of atrial fibrillation. European Heart Journal p. ehae176 (2024)
3. Gilon, C., Grégoire, J.M., Bersini, H.: Forecast of paroxysmal atrial fibrillation using a deep neural network. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–7 (2020)
4. Gilon, C., Grégoire, J.M., Mathieu, M., Carlier, S., Bersini, H.: IRIDIA-AF, a large paroxysmal atrial fibrillation long-term electrocardiogram monitoring database. Scientific Data **10**(1), 714 (2023)
5. Grégoire, J.M., Gilon, C., Carlier, S., Bersini, H.: Role of the autonomic nervous system and premature atrial contractions in short-term paroxysmal atrial fibrillation forecasting: Insights from machine learning models. Archives of Cardiovascular Diseases **115**(6), 377–387 (2022)
6. Li, Z., et al.: An early warning method for arrhythmias in long-term ECGs based on self-supervised learning and LSTM. Knowledge-Based Systems **327**, 114137 (2025)
7. Liang, Z., et al.: CTRhythm: Accurate Atrial Fibrillation Detection from Single-Lead ECG by Convolutional Neural Network and Transformer Integration. In: 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 4452–4458 (2024)
8. Moody, G.B., Mark, R.G.: A new method for detecting atrial fibrillation using RR intervals. Computers in Cardiology **10**, 227–230 (1983)
9. Rooney, S.R., et al.: Forecasting imminent atrial fibrillation in long-term electrocardiogram recordings. Journal of Electrocardiology **81**, 111–116 (2023)
10. Tzou, H.A., Lin, S.F., Chen, P.S.: Paroxysmal atrial fibrillation prediction based on morphological variant *P*-wave analysis with wideband ECG and deep learning. Computer Methods and Programs in Biomedicine **211**, 106396 (2021)
11. Venkatesh, N.P., Kumar, R.P., Neelapu, B.C., Pal, K., Sivaraman, J.: Automated atrial arrhythmia classification using 1D-CNN-BiLSTM: A deep network ensemble model. Biomedical Signal Processing and Control **97**, 106703 (2024)