A Generalizable Hybrid Deep Learning Algorithm for the Detection of Atrial Fibrillation from Diverse Electrocardiogram Data

Shai Aarons
Supervisors: Deshen Moodley, Mbithe Nzomo
University of Cape Town
Cape Town, South Africa
ARNSHA011@myuct.ac.za

ABSTRACT

In this paper, we use a hybrid deep learning methodology to detect the heart disease Atrial Fibrillation from electrocardiogram (ECG) signals. The hybrid model we use is a combination of the Convolutional Neural Network and the Bidirectional Long Short Term Memory architectures. We use four open-source ECG datasets from varying geographic origins to train the hybrid model. We performed three rounds of experiments, wherein the first of these experiments, we trained the model separately on each individual dataset. In the first round of experiments, the average accuracy is 86.2633%, and the average F1 Score is 86.5783%. For our second experiment, we train the model separately on datasets from both the United States of America (USA) and China. When testing on a separate single dataset originating from a different location to that of the data on which the model is trained, the average accuracy is 90.4269%, and the average F1 Score is 91.1418%. Finally, in the third round of experiments, we evaluate if the model, when trained on multiple different ECG datasets, can generalize to unseen ECG data from datasets of varying geographic origin. In the third experimental phase, our model has an average test stage accuracy of 89.5340% and an average F1 Score of 90.5799%.

KEYWORDS

ECG, Deep Learning, AF, Convolutional Neural Networks, Recurrent Neural Networks, Hybrid Deep Learning

1 INTRODUCTION

Atrial fibrillation (AF) is the most ubiquitous type of cardiac arrhythmia [5]. The AF class of heart disease has severe consequences such as strokes, heart failure, or even mortality [47]. The pervasiveness of AF is increasing worldwide, and its prevalence is steadily rising in developing nations [5, 8]. Of these developing nations, AF is widespread in countries such as China and Brazil [30]. Despite being a common disease affecting the elderly and other groups, AF has been demonstrated to have a high hospitalization rate, resulting in the costly use of healthcare resources [40]. Therefore, it is crucial to develop algorithms that help with the early diagnosis of this type of cardiac disease - especially with the increased usage of wearable devices and other healthcare-equipped wearables [48]. Moreover, rapid diagnostic technologies will be critical in relieving the already strained healthcare system worldwide.

The ECG is a measuring tool that reads variations in heartbeat and rhythm. ECGs record cardiac electrical activity as a signal against time by placing electrodes on various body regions, such as the chest [38]. The ECG is the primary diagnostic technique used for identifying AF [42].

Previously, knowledge-engineered algorithmic techniques or feature extraction approaches were often utilized in clinical settings to identify AF [4, 39]. But these methods frequently produce false positives, which can result in inaccurate diagnoses and inappropriately administered therapies or treatments [19]. Similarly, classical machine learning (ML) algorithms have been used for the task of AF detection. Examples of these classical techniques used for AF detection are Support Vector Machines [22] and Random Forests [6]. These classical ML techniques have been shown to overfit to the ECG training data [29]. Therefore, for the task of AF detection, we require a class of algorithms that are robust against noisy ECG data while providing accurate diagnoses. The solution is deep learning.

Deep learning has surpassed its classical forerunners thanks to its capacity to identify patterns and extract features from large amounts of raw data. [28]. This class of supervised learning models is robust against noise and, for the purpose of our study, resilient to noisy ECG data [49]. Traditional examples of deep learning models are Convolutional Neural Networks (CNNs) [33] and Recurrent Neural Networks (RNNs) [24]. Improvements to both of these aforementioned deep learning techniques have been made, such as the formation of the Long Short Term Memory (LSTM) [52] and Bidirectional LSTM (BiLSTM) [12] models. Hong et al. [16] suggest that combining CNNs and recurrent architectures into a hybrid model performs the best out of all variations of deep learning models in detecting AF from ECG data. Hybrid architectures have the advantage of being able to combine feature extraction and temporal-data-driven techniques. For our study, we opt to use a hybrid CNN-BiLSTM architecture to detect AF from ECG data.

Typically, previous works in detecting AF from ECG data perform deep learning research using ECG data bound to only a single geographic location - usually mostly American data. To the best of our knowledge, there was not yet sufficient evidence to evaluate whether deep learning models are generalizable to data from varying geographic areas. This study uses various open-source ECG datasets from both the USA and China. We perform transfer learning in an attempt to evaluate whether our model is agnostic to these physical locations and datasets when detecting AF.

This paper is laid out in the following format: firstly, Section 2 provides a brief background of previous works in detecting AF from ECG data using hybrid deep learning techniques. Next, Section 3 provides a detailed insight into the methodology we took in pursuing our research. This section includes a description of the datasets we used, an outline of the preprocessing pipeline, a description of the architecture of the deep learning model, details about how we train the model, and the evaluation metrics we use in our experiments. Section 4 reports on the results we achieved from our experimentation, and Section 5 is a discussion regarding these results.

2 BACKGROUND

A hybrid deep learning methodology is when two or more deep learning architectures are combined to, in this context, perform the classification of AF. Hybrid deep learning architectures, such as the combined CNN-BiLSTM, have the benefits of discarding feature-engineering modules and allow for feature extraction with minimal domain knowledge [45]. It is apparent from the literature that the combination of recurrent architectures and a CNN module performs impressively concerning the task of cardiovascular disease detection from ECG data [2, 32, 36].

Ivanovic et al. [17] proposed a hybrid model incorporating three CNN layers and a BiLSTM layer. The authors state that by using bidirectional LSTMs, they are able to achieve an accuracy of 89.67% in detecting AF. Oh et al. [32] demonstrated a noteworthy accuracy of 98.42% for detecting arrhythmias by using a combined structure of LSTM and CNN layers. Petmezas et al. [36] proposed a joint CNN and LSTM architecture that used the focal loss function to classify four classes of ECG rhythm types. Using the MIT-BIH AF dataset with two leads, their model achieved sensitivity and specificity scores of 97.87% and 99.29%, respectively.

While these studies indicate impressive performance, they are either trained on private datasets (such as in Ivanovic et al. [31]) or trained using only one or two datasets. This theme of a lack of diverse and open-source datasets is seen throughout the literature regarding the usage of a hybrid architecture for detecting cardiovascular diseases [2, 17, 32, 36, 50, 51]. Furthermore, many previous works have used twelve leads of ECG, placed on many parts of each patient's body, to train their models [7, 37, 45]. This has two disadvantages. Firstly, twelve leads may be unsuitable with the advent of wearable health devices and the increasing need for rapid detection. This is because wearable ECGs, such as the AliveCor or the Apple Watch, are typically only single-lead devices [18]. Secondly, as Martin et al. [23] point out, using twelve lead ECGs can produce misleadingly high testing metrics and model performance. If a deep learning model were to have clinical relevancy and efficient generalization, we should ensure the model is trained with adequately diversified data, using only a single lead.

Furthermore, across most of the literature, two sources of publicly open datasets are used to detect AF. Hong et al. [16] points out that 150 out of 191 deep learning papers in this context used open ECG datasets, and these two datasets are the most popular among them. This most commonly used dataset is the MIT-BIH Arrhythmia [26], with the second most popular being the Physionet Computing in Cardiology Challenge 2017 dataset [9]. For our study, we extend the usage of these two popular datasets with two additional open-source datasets. Our methodology takes inspiration from Zhang et al. [53], who used private ECG data to train their models but tested them on open-source data. However, we opt to use all open-source datasets instead of private data.

3 METHODOLOGY

3.1 Datasets

Most previous work in hybrid deep learning for AF detection limits their studies to one or two datasets. Much previous literature has noted that an issue with this task is a lack of data or that datasets are imbalanced [32, 36, 51]. Typically, the ML and deep learning approach to detect AF needs a significant amount of ECG data [4]. Likely, many of the previous deep learning algorithms featured in the literature may have overfitted to their limited datasets [43]. In our study, we opt to use four different open-source ECG datasets, with their respective recordings originating from varying geographic regions, namely the USA and China. We use four different datasets to mitigate overfitting and allow the model to generalize well to ECG signals with varying levels of noise and background-origin. A summary of the datasets we use for our deep learning model can be found in Table 1. We briefly outline each dataset used in our study below:

3.1.1 MIT-BIH Arrhythmia [26]. The MIT-BIH Arrhythmia dataset contains ECG data from 47 patients, all of which are around 30 minutes in duration. The ECG recordings were sampled with a frequency of 360 samples per second (360 HZ). The cardiologist-labeled ECG recordings were classified into fifteen rhythm categories - AF being one of these respective categories. The samples were recorded from two leads using 24-hour ambulatory ECG recorders. The recordings were collected from the USA at the Beth Israel Hospital in Boston (now the Beth Israel Deaconess Medical Center).

3.1.2 MIT-BIH Atrial Fibrillation (MIT-BIH AF) [27]. The MIT-BIH AF dataset contains ECG data from 23 patients, all of which are around 10 hours in duration. The ECG recordings were sampled with a frequency of 250 samples per second (250 HZ). The cardiologist-labeled ECG recordings were classified into four rhythm categories - AF being one of these respective categories. The samples were recorded from two leads using analog ambulatory ECG recorders. Like the MIT-BIH Arrhythmia dataset, the recordings were sampled in the USA at the Beth Israel Deaconess Medical Center (formerly Boston's Beth Israel Hospital).

3.1.3 Physionet Computing in Cardiology Challenge 2017 (Physionet) [9]. The Physionet dataset contains ECG data from 8,528 patients, with samples between 9 and 61 seconds in duration.

The ECG recordings were sampled with a frequency of 300 samples per second (300 HZ). The cardiologist-labeled ECG recordings were classified into four rhythm categories - AF being one of these respective categories. The data was collected with the AliveCor device, which functions as a single-lead ECG. The location from where the ECGs were sampled is undisclosed.

3.1.4 The China Physiological Signal Challenge 2018 (CPSC) [21]. The CPSC dataset contains ECG data from 6,877 patients, with samples between 6 and 60 seconds in duration. The ECG recordings were sampled with a frequency of 500 samples per second (500 HZ). The labeled ECG recordings were classified into nine rhythm categories - AF being one of these respective categories. The samples were recorded from various twelve leads ECG devices. The dataset contains ECG recordings collected from 11 different hospitals in China.

Dataset Name	**Country of Origin**	**Sampling Frequency (HZ)**	**Leads**	**Patient Count**	**Sample Length**
MIT-BIH Arrhythmia	USA	360	2	47	30 minutes
MIT-BIH AF	USA	250	2	23	10 hours
Physionet	Undisclosed	300	1	8,528	9-61 seconds
CPSC	China	500	12	6,877	6-60 seconds

Table 1: Dataset Summary

3.2 Preprocessing

Since our study includes multiple diverse datasets, it is crucial that we pay apt attention to the data preprocessing pipeline. Furthermore, preprocessing is pivotal in detecting AF using deep learning, as we have noticed that different preprocessing techniques provide improved model performance. In this section, we outline our preprocessing pipeline for dealing with multiple ECG datasets. Figure 1 is an illustration of this pipeline.

3.2.1 Lead I Extraction. The ECG can be registered as different leads depending on its different placements on the body. Leads I, II, and III can reflect changes from the frontage of the heart; chest leads, V1-V6, denote changes in ECG in the cross-section of the heart. These leads can be used in unison or alone for different purposes. Single-lead is computationally more efficient and lightweight, offering decreased training times, whereas multi-lead has a higher data dimensionality. However, as mentioned, using multiple leads may not provide relevant results and potentially demonstrate exaggerated model performance. Moreover, most wearable health devices that have ECG functionality are typically only single-lead devices [18]. For example, popular smartwatches such as the Apple Watch and the Garmin range of smartwatches are Lead I ECG devices. For these reasons, similar to Oh et al. [32] and Acharya et al. [1], we extract only a single lead from each patient's ECG recording, except we extract lead I instead of Lead II so that we can achieve healthcare wearable relevancy. This step was unnecessary for data coming from the Physionet dataset as it is a single-lead dataset containing only lead I signals.

3.2.2 R-peak Detection. The extracted single-lead signal from the above step represents an extended raw ECG signal. We will need to extract each patient's sample into smaller segments due to varying sample sizes between and within datasets. However, before we can do this, we need a means by which to traverse the sample in a manner that creates informative segments for classification. The foremost indications of the presence of AF are the absence of the P-wave [14] and/or irregular R-R intervals [29] on an ECG heartbeat reading. The heartbeats recorded by an ECG for Normal Sinus Rhythms (NSR) consist of the QRS complex, the P-wave, and the T-Wave. The maximum amplitude of the heartbeat is indicated by the R-peak, which forms part of the QRS complex. The R-R interval is the difference between two R-peaks in consecutive heartbeats [46]. To capture sufficient information in each segment, we traverse our segments based on the R-peaks of each of the single-lead signals. This ensures that every ECG segment contains at least one heartbeat. To do this, we use the Pan-Tompkins QRS detector algorithm [34] to find the R-peaks of each sample. The Pan-Tompkins algorithm performs a series of low-pass, high-pass, and derivative filters to the signal, followed by squaring the signal to amplify the QRS portion in an attempt to delineate the QRS complex. As a final step, the algorithm utilizes adaptive thresholds, locating the signals' R-peaks. We choose the Pan-Tompkins algorithm as it is the most widely used QRS detection algorithm [10]. This step was unnecessary for the MIT-BIH Arrhythmia dataset as the R-peaks are already annotated.

3.2.3 Signal Segmentation. Since we use a variety of different datasets, there are signals that have lengths that range from 6 seconds to 10 hours long. As we use a CNN-BiLSTM structure, a standard signal length must be used as input to the deep learning model. In a similar manner to Martin et al. [23] and Ghiasi et al. [11], we perform this segmentation by taking a one-second length segment before and after each R-peak. This results in segments that are two seconds in length. We found that two second-long segments provide improved accuracies in our model. Next, the associated label for each segment is extracted and stored. Occasionally, segments have a mixed rhythm label set. When this occurs, we take the following approach for labeling:

(1) If a segment contains any signals of AF, label it as 'AF.'
(2) If a segment contains no signals of AF or any other rhythm abnormalities, label it as 'NSR.'
(3) Otherwise, label the segment as 'other rhythm.'

This approach was taken to emulate a clinical setting where a physician can determine whether a patient shows signs of AF on a single segment of an ECG reading. The length of a segment can be calculated using the equation as indicated in Formula 1. This will be important for the next step in the preprocessing pipeline

$$Segment\ Length = 2 * Sampling\ Frequency \tag{1}$$

3.2.4 Signal Downsampling. In this study, we use four different datasets, each with varying sampling frequencies and, therefore, different lengths for two-second ECG segments. In order to use the segments as input to the hybrid deep learning model, it is required that each segment have the same size. Therefore, to ensure the same segment length while maintaining annotation correctness, we must downsample segments to a minimum segment length. When using datasets in unison, we find this segment length minimum by finding the minimum segment length (as described in Formula 1) of the datasets being used together. For our purposes, with reference to Table 1, the minimum segment length is equal to 500. We downsample every segment that has a length that does not match this minimum segment length. We use SciPy's[1] resample method to perform this transformation. This downsampling is performed using Fast Fourier Transforms [31] that effectively downsample segments to have lengths that match the minimum segment length while maintaining adequate information in the segment.

3.2.5 Transform for Binary Classification. In our study, we are performing a binary classification. We only require that our model learns whether or not an ECG segment demonstrates AF. Thus, we dispose of extraneous segments with rhythm annotations that do not conform to either of the 'AF' and 'NSR' class categories.

3.2.6 Segment Oversampling. Most of the datasets used in this study are imbalanced, with a high frequency of 'NSR' segments appearing in the samples. Previous works such as Oh et al. [32] suggest that using datasets such as MIT-BIH Arrhythmia in deep learning produces models that are trained on imbalanced datasets. To rectify this imbalance, we attempt to oversample the underrepresented AF segments. Oversampling has been proven to be an effective technique for handling imbalanced datasets [25]. We oversample the segments by tiling the AF data by a factor determined by Formula 2:

$$Oversample\ Factor = floor(\frac{Number\ of\ NSR\ Segments}{Number\ of\ AF\ Segments}) \quad (2)$$

3.2.7 Normalization. In order to further standardize the multiple high-frequency ECG datasets, we perform normalization on all of the segments. We normalize the segments by transforming each ECG segment to a range of between -1 and 1. This was done using a Min-Max Scaler [35] on these segments.

3.3 Network Architecture

Our proposed deep learning methodology is composed of three modules combined together to form a working hybrid model. Our model's high-level structure takes loose inspiration from Ivanovic et al. [17] as they use a hybrid CNN-BiLSTM for a similar task to ours. We use the same amount of convolutional and BiLSTM layers in our model as theirs does (with the same activation functions), except we use more learnable features in each of these layers. These differing feature sizes will be discussed in more detail in the paragraphs below. All the layers in our model follow the same order as the layers in the hybrid model pipeline proposed by Ivanovic et al. [17].

The first module uses CNN layers to perform feature extraction. This CNN module contains Max Pooling, and we use the same pooling parameters as those used by Ivanovic et al. [17], such as stride and pool size. The second module includes a BiLSTM architecture configured to extract temporal patterns and features from the CNN module output. The model proposed by Ivanovic et al. [17] uses a masking layer between their CNN module and their BiLSTM module, however, we do not use a masking layer in our architecture. Unlike the model we base ours on, we apply dropout regularization between the BiLSTM module and the final module. This final module uses a dense layer with a Sigmoid activation function to perform the binary classification. The high-level structure of the model can be found in Figure 2. We discuss the three modules in detail below:

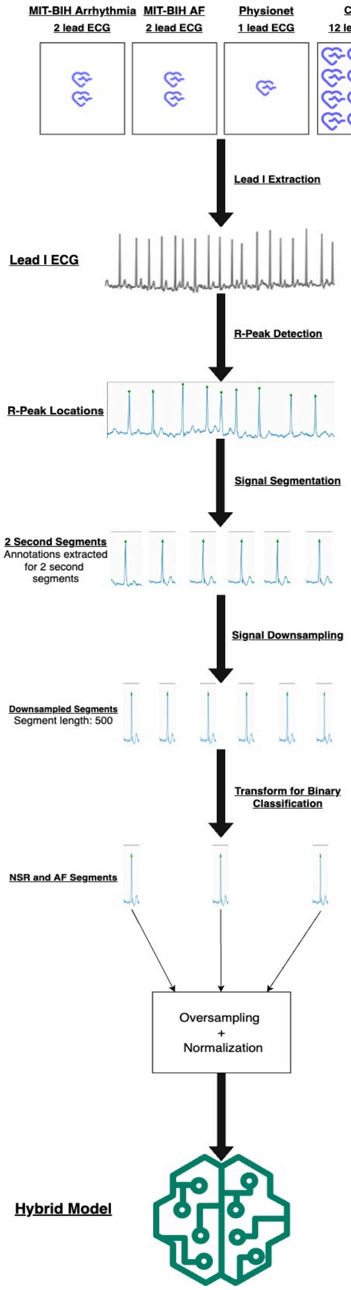

Figure 1: Preprocessing Pipeline

3.3.1 CNN Module. CNNs build on standard deep neural networks by adding extra computations for handling multi-dimensional input. A convolution is a mathematical procedure that takes a sliding window over an input space and identifies where in the input space

[1] https://scipy.org/

a pattern arises [43]. We use CNNs as they are less sensitive to noise and effectively extract patterns from the noise [32].

In our model, the input is the size of the segment length, which depends on the combination of datasets being used. Each segment is fed into two successive one-dimensional convolutional layers, both of which use the rectified linear unit (ReLu) activation function. The first convolutional layer outputs 128 learned features, using a kernel size of 5, whereas the first convolutional layer in the model proposed by Ivanovic et al. [17] outputs 60 learned features. In our model, these learned filters are passed into the second convolutional layer with a filter size of 256 and a kernel size of 3. Ivanovic et al. [17] use a filter size of 80 in their second convolutional layer. After the first two convolutional layers in our architecture, Max Pooling is applied to the output of the second convolution layer, with a stride of 2 and pool size of 2. We then apply a dropout regularization of 0.2 to reduce overfitting, similar to Ivanovic et al. [17], who use a dropout rate of 0.05 here. The next layer in our model is an additional convolution layer with 512 filters and a kernel size of 3, similarly using the ReLu activation function. This third convolutional layer is different from the model we base ours on since theirs uses 128 filters instead of 512. From our third convolutional layer's output, max-pooling is applied once again with a kernel size of 2 and a stride of 2. Once again, dropout is used with a rate of 0.2 to improve generalization. Ivanovic et al. [17] used a lower dropout rate of 0.15. The output is then fed into the BiLSTM module.

3.3.2 BiLSTM Module. Vanilla RNNs suffer from the vanishing gradient problem where the error calculated during training has minimal effect as it gets further backpropagated through the RNN [15]. The LSTM is a structure that seeks to rectify this issue of short-term memory [52]. The LSTM architecture incorporates cells/blocks with gates that function as activation functions and are used for learning temporal data. Using these gates, the LSTM cell can maintain a cell memory and opt for what memory from previous cells is preserved or forgotten. The gates include the forget gate, the input gate, and the output gate [13]. The BiLSTM is a similar structure to that of the LSTM; however, it incorporates not only previous information but also future information from future time steps into the learning process [51]. We found that using a BiLSTM allows for higher performance concerning our AF classification task.

The output from the CNN module of our model is used as input for our BiLSTM model. We use a BiLSTM layer with 256 hidden units, containing 128 units in each direction, whereas Ivanovic et al. [17] use 50 units in each direction. The BiLSTM portion of the model uses the tanh activation function. We found that increasing the hidden units combined with a high dropout rate following the LSTM layer provided improved model performance and more generalizable results. Therefore, we apply a dropout of 0.4 on the output of the BiLSTM layer. The result is flattened and fed into the final layer for classification.

3.3.3 Output Module and Loss Function. The output from the BiLSTM layer is fed directly into a dense layer with 1 unit for learning. This dense layer functions as our output layer. Since we are performing binary classification, we use the Sigmoid activation function for this layer.

Adapted from Petmezas et al. [36], we opt to use a binary focal loss function for our architecture. The focal loss function has proven to be effective for classification tasks that use imbalanced datasets [20]. Despite already oversampling our data, we found that using focal loss in conjunction with the CNN-BiLSTM structure improved model learning and performance.

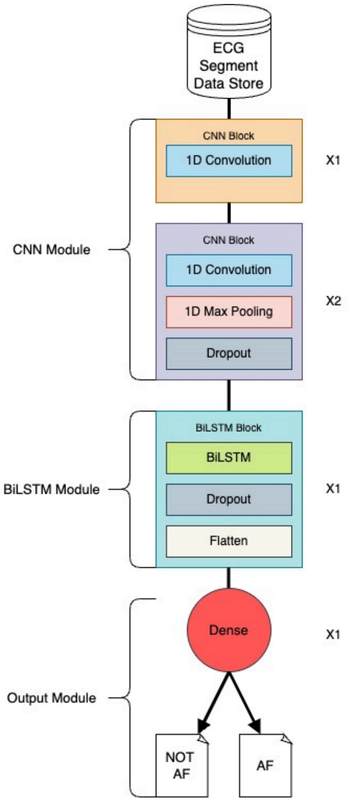

Figure 2: Deep Learning Network Structure

3.4 Data Splitting Methodology

For each iteration of training our model (See Section 4), we are required to split the combined multiple datasets into suitable train, validation and test splits. To avoid bias, we opt to use a data-splitting method that separates the data on a patient basis. This means that all segments originating from a specific patient can occur in only one of the train, validation, and test datasets. As Martin et al. [23] points out, this unbiased patient splitting method makes it more likely that the test findings will accurately reflect true model performance when attempting to classify the ECGS from new and unseen patients. In addition, this method ensures a model with less inflated results than one trained by arbitrarily splitting an integrated dataset. Figure 3 shows our data splitting methodology in pictorial form.

When training on only one dataset, we choose to perform a splitting methodology that uses 80% of the patients in each dataset for training the model, 10% for validating the model, and 10% for testing the model. This data-splitting technique is illustrated in the top portion of Figure 3. The validation dataset assisted us in fine-tuning the model. We also use the validation set to reduce the learning rate of the model whenever the validation loss plateaus while learning. The test set is used to evaluate model performance on unseen data.

However, when we train the model on multiple datasets using our 'warm-start' approach (See Section 4), we follow the procedure as outlined in the bottom portion of Figure 3. In the initial round of training, multiple datasets are split into 90% for training and 10% for validation. The validation dataset is used here for similar purposes of fine-tuning and reducing the learning rate. The model is only then tested on a hold-out dataset after fine-tuning the model using that hold-out dataset for the second round of training. The hold-out dataset is split similarly to the single-dataset splitting method, where 80% of the patients from that dataset are used for training, 10% for validation, and 10% for testing. The validation split of the hold-out dataset is again used for fine-tuning, and the test portion is used for our final test stage metric reporting.

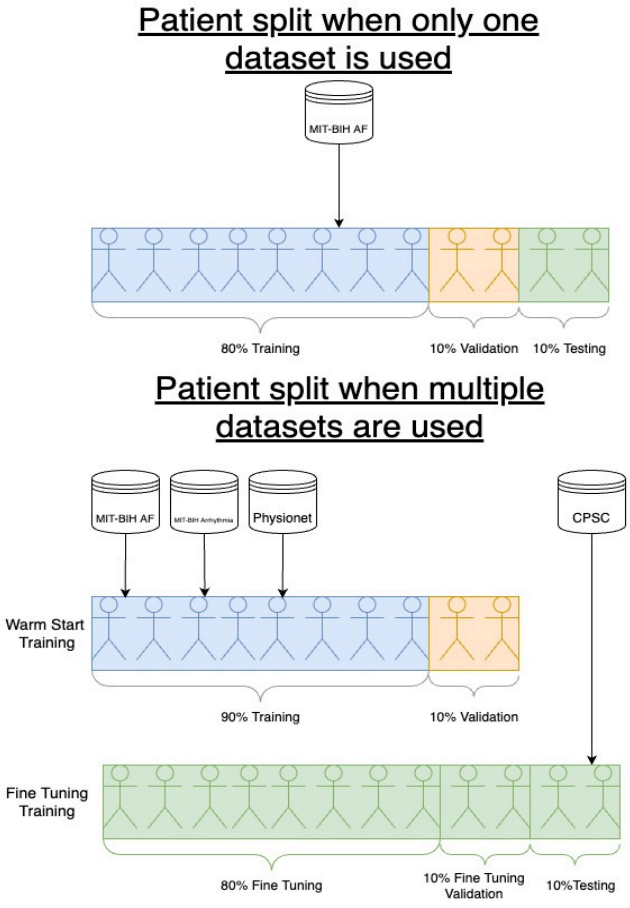

Figure 3: Data Splitting Methodology

3.5 Model Training

For each experiment, we train the model using a set of hyperparameters found through a grid search. The hyperparameters we searched over and our selected hyperparameters for training can be found in Table 2. We found that our model performs with the highest accuracy when using a batch size of 32. We train each model with an initial learning rate of 0.001. When the 'warm start' (See Section 4) approach is employed, we fine-tune the model for a smaller number of epochs on a new dataset with an initial learning rate of 0.0001 to avoid biasing the model to the new dataset. As previously mentioned, we found that training on segment lengths of 2 seconds provided improved model performance. We opt to use the Adam optimizer and a learning rate scheduler that decrements the learning rate whenever learning plateaus. We additionally use early stopping to ensure that the model does not overfit.

Hyperparameter	Options	Selected Options
Batch Size	32, 64, 128, 256	32
Initial Learning Rate	0.01, 0.001, 0.005, 0.0001	0.001
Segment Size	0.4, 0.5, 1, 2, 4, 6	2
Maximum Epochs	40, 60, 100, 150, 250	150
Loss Function	Binary Focal, Binary Cross Entropy	Binary Focal
Optimizer	Stochastic Gradient Descent, Adam	Adam

Table 2: Selected Hyperparameter Options for Training

3.6 Evaluation Metrics

It is helpful to refer to the following parameters when evaluating the performance of an ML model when detecting AF: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). We can use these values to create a confusion matrix, which allows for interpretation of results in a non-biased fashion. The confusion matrix is set up as follows:

$$\begin{pmatrix} TN & FN \\ FP & TP \end{pmatrix}$$

The first metric we use for model evaluation is accuracy. Accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Another set of measures we opt to use are sensitivity (recall) and precision. These depict the algorithm's ability to distinguish between different task outcomes. The sensitivity score has significant importance for biomedical applications. The formula for these metrics is given below:

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

An additional suitable statistic of model performance is the F1 Score. We use the F1 Score as it is the harmonic mean of sensitivity and precision and summarizes the two competing metrics of accuracy and sensitivity. The F1 Score allows us to compare different algorithm's performance using the following formula:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

4 EXPERIMENTS AND RESULTS

4.1 Experiment Setup

The experimentation for our CNN-BiLSTM hybrid model performance has a three-phased approach. The experiments build on one another by having each subsequent investigation expand the size of data used to train our hybrid deep learning model while having each experiment serve different objectives. The first experiment involves training and testing the model on only a single dataset. We perform this first investigation, training the model separately on each dataset. This experiment aims to demonstrate how our hybrid model performs using only single datasets. The results from this experiment serve as a means of comparison for the generalizability of our model. We compare the evaluation metrics of the first round of experiments to those achieved in the second and third rounds of experiments.

In our second experiment, we train on datasets that originate from the USA and then use the CPSC dataset as a hold-out dataset for fine-tuning and testing. We also do this in reverse order, where we train on the dataset originating from China and test on a single dataset from the USA (MIT-BIH Arrhythmia). Training on the first regional dataset allows the model to attain a ‘warm-start’ [3]. This second round of tests evaluates the model’s ability to generalize predictions to different geographic locations when having been pre-trained on data from other regions.

For the third experiment, we build on the second experiment by having four separate iterations of training of the model - these iterations function as an adjusted cross-validation approach. In each iteration, three (out of the four) datasets are used for training the model, and one dataset is held out of the training. We then fine-tune each trained model by further training on the hold-out dataset. Training on the first three datasets allows the model to attain a ‘warm-start.’ Each trained model is then tested on the test portion of the hold-out dataset. The results from this bootstrapped/transfer learning approach are compared with those achieved in the first and second rounds of experiments. By comparing the results achieved from the first experiment with the results achieved when the corresponding dataset is hold-out and tested in the third experiment, we can demonstrate that by training on a larger integrated dataset, we are able to produce more generalizable and higher-performing models for detecting AF from ECG signals.

4.2 Implementation Details

For each of the experiments, we train the model on Google’s Colab Pro Plus[2]. The Colab platform allows for 51GB of RAM usage with an NVIDIA Tesla P100 PCIe 16GB GPU. We coded our experimental platform and the hybrid deep learning model using TensorFlow’s Keras. Our experimental platform allows us to select different datasets to be used for training and testing, as well as allow for hyperparameter adjustments and preprocessing method selection. All libraries used in the experimental platform, such as Keras[3], NumPy[4], and SciPy[5], are open-source.

4.3 Experiment One

In this experiment, we train the model using only one dataset at a time. We run separate training rounds for each dataset, with a newly generated model containing newly generated random weights. We use the same hyperparameters for the model across these experiments, these include batch size and initial learning rate. The highest test accuracy achieved with this round of experimentation was 90.5183% using the MIT-BIH AF dataset to train the model. Another noteworthy test accuracy achieved was 86.4878% when we used the CPSC dataset for training. The confusion matrices for the results of this round of experiments can be found in comparison with the corresponding confusion matrices in the second and third experiments in Figures 4 and 8, respectively. The average test accuracy for this experiment was 86.2633% across all datasets. All the evaluation metrics for unseen data in these training rounds can be found in Table 3. The average of these evaluation metrics can be found at the bottom of Table 3.

Dataset	**Accuracy (%)**	**Sensitivity (%)**	**Precision (%)**	**F1 Score (%)**
MIT-BIH Arrhythmia	83.4250	68.9413	99.8802	81.5758
MIT-BIH AF	90.5183	88.8424	93.2312	90.8789
Physionet	84.6219	81.0447	89.3750	85.0006
CPSC	86.4878	81.5289	97.6347	88.8579
Average Test Metrics	**86.2633**	**80.0893**	**95.0303**	**86.5783**

Table 3: Evaluation Metrics for Experiment One.

4.4 Experiment Two

We perform two additional experiments in our second round of experimentation. In the first test for these experiments, we train on American datasets (MIT-BIH AF and MIT-BIH Arrhythmia), and fine-tune/test on the dataset from China (CPSC). Here the accuracy achieved is 94.7313%. We perform a similar test in reverse order, where we have an initial round of training on the Chinese dataset and then fine-tune and test on a single American dataset - MIT-BIH Arrhythmia. We choose MIT-BIH Arrhythmia as it has the lowest sensitivity score in the first set of experiments. The recorded test stage accuracy for this second test is 86.1224%, and the test sensitivity is 83.9905%. The in-depth results for this round of experiments can be seen in Table 4. Table 5 serves as a summary table that compares all the evaluation metrics from the first experiment to when the same corresponding dataset is used as a hold-out dataset in this second experiment. We compare the confusion matrices of these tests with those of the first experiments that use the same datasets as the datasets that are hold-out in this experiment. This confusion matrix comparison can be found in Figure 4.

[2] https://colab.research.google.com/signup

[3] https://keras.io/

[4] https://numpy.org/

[5] https://scipy.org/

Train Datasets	Testing Dataset	Accuracy (%)	Sensitivity (%)	Precision (%)	F1 Score (%)
CPSC	MIT-BIH Arrhythmia	86.1224	83.9905	89.2583	86.5443
MIT-BIH Arrhythmia, MIT-BIH AF	CPSC	94.7313	92.8242	98.8433	95.7393
Average Test Metrics		**90.4269**	**88.4074**	**94.0508**	**91.1418**

Table 4: Evaluation Metrics for Experiment Two.

	Accuracy (%)		Sensitivity (%)		Precision (%)		F1 Score (%)	
Testing Dataset	**E1**	**E2**	**E1**	**E2**	**E1**	**E2**	**E1**	**E2**
MIT-BIH Arrhythmia	83.43	86.12	68.94	83.99	99.88	89.23	81.58	86.54
CPSC	86.49	94.73	81.53	92.82	97.64	98.84	88.86	95.74
Average Test Metrics	**86.26**	**90.43**	**80.09**	**88.41**	**95.03**	**94.05**	**86.58**	**91.14**

Table 5: Comparison of Experiment One (E1) and Experiment Two (E2) Evaluation Metrics.

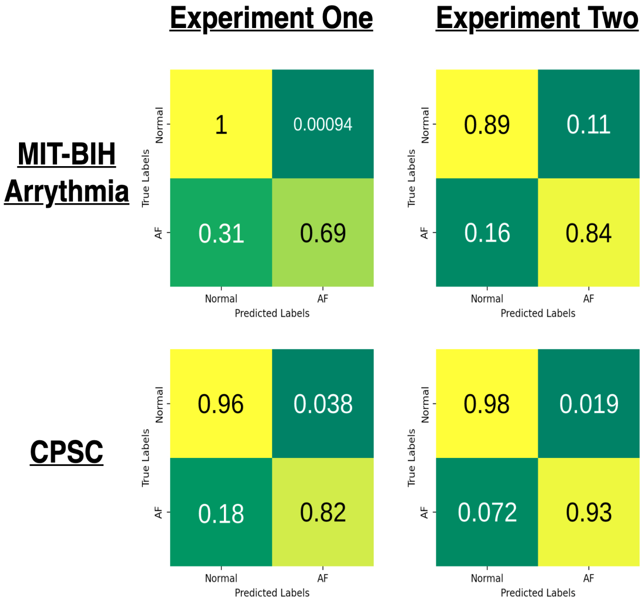

Figure 4: Confusion Matrix Comparison for Experiments One and Two.

4.5 Experiment Three

In this final round of experiments, we train the model using three datasets while keeping a hold-out dataset for testing. In each 'fold' of this experiment, the model is trained on three datasets and then fine-tuned on the train portion of the hold-out dataset. We note that when MIT-BIH Arrhythmia is used for training with a 'warm start' from the other datasets, its test stage accuracy is 89.1858%, and the test stage sensitivity is 89.7976%. Similarly, after pre-training the model and testing on the MIT-BIH AF, the test stage accuracy is 82.8237% and detects true positive AF segments with approximately 99% accuracy. When the Physionet dataset is hold-out, the test accuracy achieved by the model is 88.0667%. Finally, the iteration that saw the highest accuracy of 98.0598% was when the CPSC was used as the hold-out dataset. The CPSC also saw the highest sensitivity of 98.0523% and precision of 99.0311%.

Table 6 shows all the evaluation metrics for all iterations of this experiment. The bottom of Table 6 demonstrates the average evaluation metrics across all four of the datasets used for hold-out testing. Table 7 serves as a summary table that compares all the evaluation metrics from the first experiment to when the same dataset is used as a hold-out dataset in this third experiment. For a visual comparison, Figure 5 is a bar chart demonstrating the test stage accuracy of each dataset used in each round of experiments. Similarly, the test stage sensitivity of each dataset in all of the rounds of experiments is shown in Figure 6. The bar charts for test stage precision and F1 Score can be found in Figures 9 and 10 in Appendix A. We include a final bar chart in Figure 7 to illustrate the average evaluation metrics of our hybrid model performance across all phases of experimentation. Figure 8 depicts the confusion matrices for experiment three in comparison with the corresponding confusion matrices from experiment one.

Train Datasets	Testing Dataset	Accuracy (%)	Sensitivity (%)	Precision (%)	F1 Score (%)
MIT-BIH AF, Physionet, CPSC	MIT-BIH Arrhythmia	89.1858	89.7976	90.4138	90.1047
MIT-BIH Arrhythmia, Physionet, CPSC	MIT-BIH AF	82.8237	98.7602	75.6977	85.7046
MIT-BIH Arrhythmia, MIT-BIH-AF, CPSC	Physionet	88.0667	81.8624	95.0644	87.9709
MIT-BIH Arrhythmia, MIT-BIH-AF, Physionet	CPSC	98.0598	98.0523	99.0311	98.5392
Average Test Metrics		**89.5340**	**92.1181**	**90.0518**	**90.5799**

Table 6: Evaluation Metrics for Experiment Three

	Accuracy (%)		Sensitivity (%)		Precision (%)		F1 Score (%)	
Testing Dataset	**E1**	**E3**	**E1**	**E3**	**E1**	**E3**	**E1**	**E3**
MIT-BIH Arrhythmia	83.43	89.19	68.94	89.80	99.88	90.41	81.58	90.10
MIT-BIH AF	90.52	82.82	88.84	98.76	93.23	75.70	90.88	85.70
Physionet	84.62	88.07	81.05	81.86	89.38	95.06	85.00	87.97
CPSC	86.49	98.06	81.53	98.05	97.64	99.03	88.86	98.54
Average Test Metrics	**86.26**	**89.53**	**80.09**	**92.12**	**95.03**	**90.05**	**86.58**	**90.58**

Table 7: Comparison of Experiment One (E1) and Experiment Three (E3) Evaluation Metrics

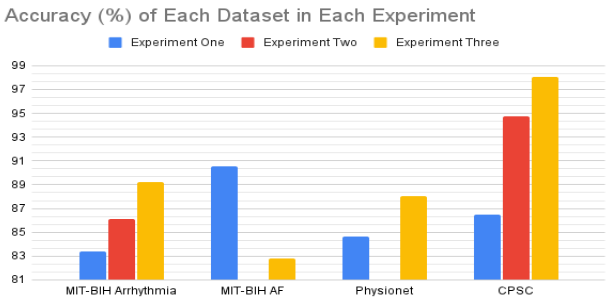

Figure 5: Bar Chart Comparing Accuracy of Each Dataset Used in Each Experiment

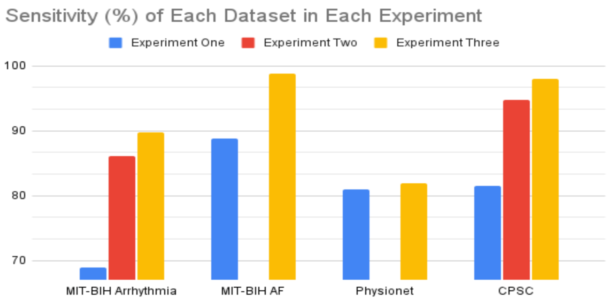

Figure 6: Bar Chart Comparing Sensitivity of Each Dataset Used in Each Experiment

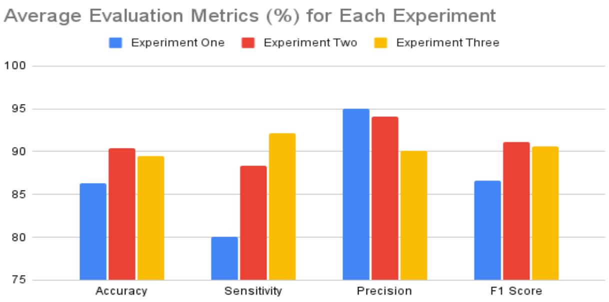

Figure 7: Bar Chart Comparing Average Evaluation Metrics in Each Experiment

5 DISCUSSION

From the first round of experiments, where we train on individual datasets, we attain an average accuracy of 86.2633%. The MIT-BIH AF dataset performed particularly well in this round of training with an accuracy of 90.5183%. This higher accuracy is likely attributed to the fact that the MIT-BIH AF dataset produces significantly more ECG segments than the other datasets, with a more balanced representation of AF segments. For two-second segments, in a typical round of training, the proportion of NSR segments to unique AF segments in the training split is approximately 1.05 when using the MIT-BIH AF dataset. Whereas with a dataset such as MIT-BIH Arrhythmia, the proportion of NSR segments to unique AF segments in the training split is approximately 6.4. This likely explains the lower sensitivity of 68.9413% when we train on only the MIT-BIH Arrhythmia in the first round of experiments.

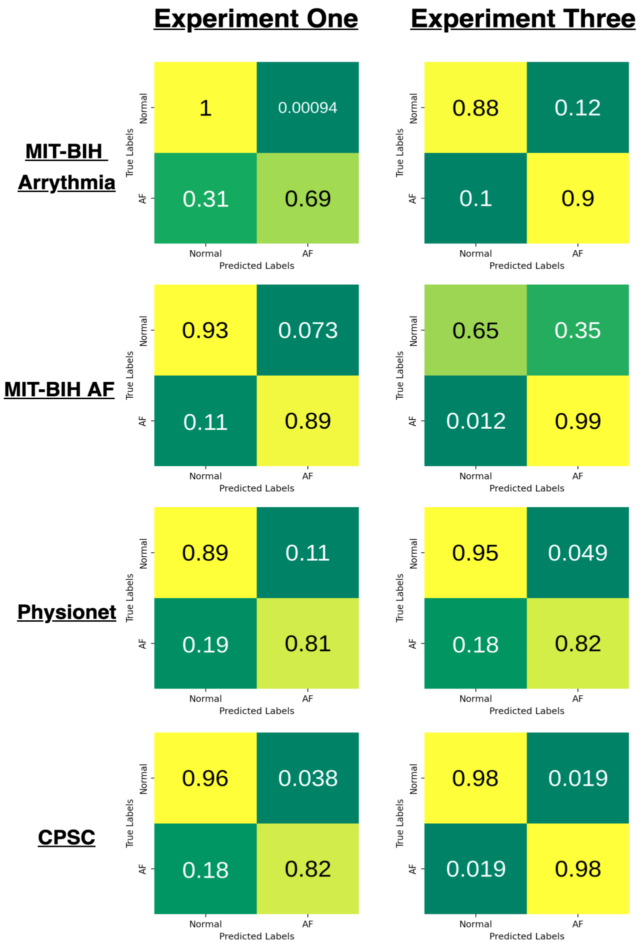

Figure 8: Confusion Matrix Comparison for Experiments One and Three

This theme of class imbalance influencing test accuracy and sensitivity also explains the decrease in accuracy when MIT-BIH AF is used as the hold-out dataset compared to its accuracy in experiment one. When the MIT-BIH AF dataset is used as a hold-out dataset, the model is first trained on a selection of imbalanced datasets (MIT-BIH Arrhythmia, Physionet, and CPSC). Nevertheless, the sensitivity when testing on the MIT-BIH AF dataset increased from 88.8424% to 98.7602%. This indicates that its ability to detect AF increased significantly, despite its precision decreasing in this round of training.

Excluding the MIT-BIH AF dataset, all other datasets saw improvements in accuracy and sensitivity when used as test sets in the third experiment. The MIT-BIH Arrhythmia dataset had an accuracy increase from 83.4250% to 89.1858% when the model was trained on the three other datasets. It also receives an accuracy and sensitivity increase (86.1224% and 83.9905%, respectively) when used as a test set for the model trained on only the CPSC dataset in the second round of experiments. The MIT-BIH Arrhythmia has a high level of class imbalance, and in the second and third experiments, this class balance is offset during training by the other more balanced datasets, such as MIT-BIH AF - hence the performance improvement.

Notably, in the first round of training, when the CPSC dataset is used to train the model, the accuracy is 86.4878%, however, when the model is trained on only data originating from the USA, the accuracy and sensitivity when testing on the CPSC dataset are 94.7313% and 92.8242%, respectively. When the model is trained on the three other datasets and tested on the CPSC dataset, the accuracy is 98.0598%, and the sensitivity is 98.0523%. Similarly, the accuracy when experimenting with the Physionet dataset also saw an increase in experiment one of 84.6219% to an accuracy of 88.0667% in the third experiment. In the second experiment, the average accuracy, sensitivity, and F1 Scores saw improvements compared to the average performance metrics across training iterations in the first phase. Besides the MIT-BIH AF dataset, the F1 Scores, when using datasets as hold-out testing datasets in the third round of experiments, all saw improvements compared to their respective F1 Scores in the first round of experiments. The confusion matrix comparisons in Figures 4 and 8 illustrate that the model's ability to detect true AF segments increases for every dataset when having been pre-trained on other ECG data. The performance increase in the second and third rounds of tests indicates our model's ability to generalize to unseen data when trained on balanced datasets

We have presented a deep learning architecture that is based loosely upon the hybrid CNN-BiLSTM proposed by Ivanonic et al. [17] but is significantly different in terms of structure, layers, and features from other hybrid convolutional and recurrent architectures in previous literature [2, 32, 36, 51]. We note that as our model receives more data, it can generalize sufficiently across geographic regions. When we train primarily on data from the USA (MIT-BIH AF and MIT-BIH Arrhythmia with Physionet included in the third phase) and hold out the dataset originating from China (CPSC), the model can generalize well to the Chinese dataset with accuracies of 98.0598% and 94.7313%. When the model is trained using only Chinese data (CPSC), the model can make predictions with an accuracy of 86.1224% on data from the USA. It is evident from these results that our model demonstrates agnosticism towards the geographic origin of data or ECG recording devices. To the best of our knowledge, this deep learning model geographic location agnosticism has not been demonstrated in prior works in the literature regarding AF detection.

We use more datasets in unison than is seen in most of the literature concering AF detection using hybrid models [2, 17, 32, 36, 51]. Our model is able to learn features from these multiple datasets and perform with comparable accuracies to those previous studies using hybrid methodologies to detect AF. For example, our model's average accuracy in the second round of experiments is 90.4269% and 89.5340% in the third round of experiments. Anderson et al.'s [2] hybrid CNN-LSTM model achieved an accuracy of 89.30% using three open-source datasets: MIT-BIH Arrhythmia, MIT-BIH AF, and MIT-BIH NSR [44]. Ivanovic et al.'s [17] hybrid CNN-BiLSTM model demonstrated an accuracy of 88.28% using a private ECG dataset.

Moreover, our model, unlike many hybrid models seen in previous work [7, 36, 51], uses only a single lead for classification. We use only lead I ECG signals as this is the lead used in smartwatches and ECG wearables. Zhang et al. [53] train their hybrid model on private single-lead wearable ECG data, and when they test their model on the MIT-BIH AF dataset, their sensitivity score is 96.46%, whereas in our third experiment, when testing on the same MIT-BIH AF dataset the sensitivity is 98.0523%. Zhang et al. [53], like other previous studies [36, 41, 45], use filters and wavelet transforms to remove baseline wander and power-line interference from the noisy ECG sequences. We differentiate ourselves by using only raw lead I ECG segments for classification.

6 CONCLUSION

The key contributions of this work include the presented hybrid model that can learn from raw, single-lead ECG data split into testing, training, and validation datasets on a patient level - thus delivering a realistic and reproducible model performance for use in clinical settings. This performance is reflected in the average model accuracy of 90.4269% and sensitivity of 88.4074% in the second round of experiments and 89.5340% and 92.1181% in the third round of experiments. We have demonstrated that by training on sufficiently balanced segments of NSR and AF, the model is able to detect and classify AF from ECG data sufficiently. Our model is generalizable across the geographic bounds of China and the USA and tends to improve performance when initially trained on data from other regions than that on which it is tested. On average, our model, when trained on multiple datasets, performs better than similar hybrid models seen in the literature. Overall our hybrid CNN-BiLSTM model can generalize well to unseen ECG data.

In future work, there is an opportunity to allow for more inclusivity of ECG datasets from other geographical regions not featured in this study, such as ECG data from Africa, Europe, or South America. This will allow for further testing of generalizability to geographic regions. In the context of our study, we aim for generalization and efficiency, hence limiting our analysis to raw single-lead ECG signals; however, future works could introduce additional preprocessing steps such as filters and discrete wavelet transforms to increase model performance. Ultimately, there is a limited amount of open-source ECG data labeled by board-certified cardiologists. We encourage hospitals to make their signal data available for public use to improve clinical treatments, diagnosis, and detection.

7 ACKNOWLEDGMENTS

We thank Deshen Moodley and Mbithe Nzomo, who guided us in completing this research. We would also like to thank the Center for Artificial Intelligence Research (CAIR) for allowing us to use their hardware infrastructure.

REFERENCES

[1] U. Rajendra Acharya, Hamido Fujita, Oh Shu Lih, Yuki Hagiwara, Jen Hong Tan, and Muhammad Adam. 2017. Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network. *Information Sciences* 405 (2017), 81–90. https://doi.org/10.1016/j.ins.2017.04.012

[2] Rasmus S. Andersen, Abdolrahman Peimankar, and Sadasivan Puthusserypady. 2019. A deep learning approach for real-time detection of atrial fibrillation. *Expert Systems with Applications* 115 (2019), 465–473. https://doi.org/10.1016/j.eswa.2018.08.011

[3] Jordan T. Ash and Ryan P. Adams. 2019. On the Difficulty of Warm-Starting Neural Network Training. *CoRR* abs/1910.08475 (2019). arXiv:1910.08475 http://arxiv.org/abs/1910.08475

[4] Raymond Bond, Dewar Finlay, Salah Shafiq Al-Zaiti, and Peter Macfarlane. 2021. Machine learning with electrocardiograms: A call for guidelines and best practices for 'stress testing' algorithms. *Journal of Electrocardiology* 69 (2021), 1–6. https://doi.org/10.1016/j.jelectrocard.2021.07.003

[5] Bianca J. J. M. Brundel, Xun Ai, Mellanie True Hills, Myrthe F. Kuipers, Gregory Y. H. Lip, and Natasja M. S. de Groot. 2022. Atrial fibrillation. *Nature Reviews Disease Primers* 8, 1 (April 2022). https://doi.org/10.1038/s41572-022-00347-9

[6] Ricardo Buettner and Marc Schunter. 2019. Efficient machine learning based detection of heart disease. In *2019 IEEE International Conference on E-health Networking, Application Services (HealthCom)*. 1–6. https://doi.org/10.1109/HealthCom46333.2019.9009429

[7] Tsai-Min Chen, Chih-Han Huang, Edward S.C. Shih, Yu-Feng Hu, and Ming-Jing Hwang. 2020. Detection and Classification of Cardiac Arrhythmias by a Challenge-Best Deep Learning Neural Network Model. *iScience* 23, 3 (March 2020), 100886. https://doi.org/10.1016/j.isci.2020.100886

[8] Sumeet S. Chugh, Gregory A. Roth, Richard F. Gillum, and George A. Mensah. 2014. Global Burden of Atrial Fibrillation in Developed and Developing Nations. *Global Heart* 9, 1 (March 2014), 113. https://doi.org/10.1016/j.gheart.2014.01.004

[9] Gari D Clifford, Chengyu Liu, Benjamin Moody, Li-wei H. Lehman, Ikaro Silva, Qiao Li, A E Johnson, and Roger G. Mark. 2017. AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*. 1–4. https://doi.org/10.22489/CinC.2017.065-469

[10] M. A. Z. Fariha, R. Ikeura, S. Hayakawa, and S. Tsutsumi. 2020. Analysis of Pan-Tompkins Algorithm Performance with Noisy ECG Signals. *Journal of Physics: Conference Series* 1532, 1 (jun 2020), 012022. https://doi.org/10.1088/1742-6596/1532/1/012022

[11] Shadi Ghiasi, Mostafa Abdollahpur, Nasimalsadat Madani, Kamran Kiani, and Ali Ghaffari. 2017. Atrial fibrillation detection using feature based algorithm and deep convolutional neural network. In *2017 Computing in Cardiology (CinC)*. 1–4. https://doi.org/10.22489/CinC.2017.159-327

[12] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.

[13] Klaus Greff, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2017. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems* 28, 10 (2017), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

[14] Yuki Hagiwara, Hamido Fujita, Shu Lih Oh, Jen Hong Tan, Ru San Tan, Edward J Ciaccio, and U Rajendra Acharya. 2018. Computer-aided diagnosis of atrial fibrillation based on ECG Signals: A review. *Information Sciences* 467 (2018), 99–114. https://doi.org/10.1016/j.ins.2018.07.063

[15] Sepp Hochreiter. 1998. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06, 02 (April 1998), 107–116. https://doi.org/10.1142/s0218488598000094

[16] Shenda Hong, Yuxi Zhou, Junyuan Shang, Cao Xiao, and Jimeng Sun. 2020. Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review. *Computers in Biology and Medicine* 122 (2020), 103801. https://doi.org/10.1016/j.compbiomed.2020.103801

[17] Marija D. Ivanovic, Vladimir Atanasoski, Alexei Shvilkin, Ljupco Hadzievski, and Aleksandra Maluckov. 2019. Deep Learning Approach for Highly Specific Atrial Fibrillation and Flutter Detection based on RR Intervals. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 1780–1783. https://doi.org/10.1109/EMBC.2019.8856806

[18] Paola Kamga, Rasik Mostafa, and Saba Zafar. 2022. The Use of Wearable ECG Devices in the Clinical Setting: a Review. *Current Emergency and Hospital Medicine Reports* 10, 3 (June 2022), 67–72. https://doi.org/10.1007/s40138-022-00248-x

[19] June-Goo Lee, Sanghoon Jun, Young-Won Cho, Hyunna Lee, Guk Bae Kim, Joon Beom Seo, and Namkug Kim. 2017. Deep Learning in Medical Imaging: General Overview. *Korean Journal of Radiology* 18, 4 (2017), 570. https://doi.org/10.3348/kjr.2017.18.4.570

[20] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. *CoRR* abs/1708.02002 (2017). arXiv:1708.02002 http://arxiv.org/abs/1708.02002

[21] Feifei Liu, Chengyu Liu, Lina Zhao, Xiangyu Zhang, Xiaoling Wu, Xiaoyan Xu, Yulin Liu, Caiyun Ma, Shoushui Wei, Zhiqiang He, et al. 2018. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *Journal of Medical Imaging and Health Informatics* 8, 7 (2018), 1368–1373.

[22] Na Liu, Muyi Sun, Ludi Wang, Wei Zhou, Hao Dang, and Xiaoguang Zhou. 2018. A support vector machine approach for AF classification from a short single-lead ECG recording. *Physiological Measurement* 39, 6 (jun 2018), 064004. https://doi.org/10.1088/1361-6579/aac7aa

[23] Harold Martin, Walter Izquierdo, Mercedes Cabrerizo, Anastasio Cabrera, and Malek Adjouadi. 2021. Near real-time single-beat myocardial infarction detection from single-lead electrocardiogram using Long Short-Term Memory Neural Network. *Biomedical Signal Processing and Control* 68 (2021), 102683. https://doi.org/10.1016/j.bspc.2021.102683

[24] Larry R Medsker and LC Jain. 2001. Recurrent neural networks. *Design and Applications* 5 (2001), 64–67.

[25] Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. 2020. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)*. IEEE, 243–248.

[26] George B Moody and Roger G Mark. 1992. MIT-BIH Arrhythmia Database. https://doi.org/10.13026/C2F305

[27] George B Moody and Roger G Mark. 1992. MIT-BIH Atrial Fibrillation Database. https://doi.org/10.13026/C2MW2D

[28] Ruihui Mu and Xiaoqin Zeng and. 2019. A Review of Deep Learning Research. *KSII Transactions on Internet and Information Systems* 13, 4 (April 2019), 1738–1764. https://doi.org/10.3837/tiis.2019.04.001

[29] Fatma Murat, Ferhat Sadak, Ozal Yildirim, Muhammed Talo, Ender Murat, Murat Karabatak, Yakup Demir, Ru-San Tan, and U. Rajendra Acharya. 2021. Review of Deep Learning-Based Atrial Fibrillation Detection Studies. *International Journal of Environmental Research and Public Health* 18, 21 (2021). https://doi.org/10.3390/ijerph182111302

[30] Tu N. Nguyen, Sarah N. Hilmer, and Robert G. Cumming. 2013. Review of epidemiology and management of atrial fibrillation in developing countries. *International Journal of Cardiology* 167, 6 (Sept. 2013), 2412–2420. https://doi.org/10.1016/j.ijcard.2013.01.184

[31] Henri J Nussbaumer. 1981. The fast Fourier transform. In *Fast Fourier Transform and Convolution Algorithms*. Springer, 80–111.

[32] Shu Lih Oh, Eddie Y.K. Ng, Ru San Tan, and U. Rajendra Acharya. 2018. Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats. *Computers in Biology and Medicine* 102 (2018), 278–287. https://doi.org/10.1016/j.compbiomed.2018.06.002

[33] Keiron O'Shea and Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. https://doi.org/10.48550/ARXIV.1511.08458

[34] Jiapu Pan and Willis J. Tompkins. 1985. A Real-Time QRS Detection Algorithm. *IEEE Transactions on Biomedical Engineering* BME-32, 3 (1985), 230–236. https://doi.org/10.1109/TBME.1985.325532

[35] S Patro and Kishore Kumar Sahu. 2015. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462* (2015).

[36] Georgios Petmezas, Kostas Haris, Leandros Stefanopoulos, Vassilis Kilintzis, Andreas Tzavelis, John A Rogers, Aggelos K Katsaggelos, and Nicos Maglaveras. 2021. Automated Atrial Fibrillation Detection using a Hybrid CNN-LSTM Network on Imbalanced ECG Datasets. *Biomedical Signal Processing and Control* 63 (2021), 102194. https://doi.org/10.1016/j.bspc.2020.102194

[37] Antônio H. Ribeiro, Manoel Horta Ribeiro, Gabriela M. M. Paixão, Derick M. Oliveira, Paulo R. Gomes, Jéssica A. Canazart, Milton P. S. Ferreira, Carl R. Andersson, Peter W. Macfarlane, Wagner Meira, Thomas B. Schön, and Antonio Luiz P. Ribeiro. 2020. Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nature Communications* 11, 1 (April 2020). https://doi.org/10.1038/s41467-020-15432-4

[38] Beanbonyka Rim, Nak-Jun Sung, Sedong Min, and Min Hong. 2020. Deep Learning in Physiological Signal Data: A Survey. *Sensors* 20, 4 (2020). https://doi.org/10.3390/s20040969

[39] Santanu Sahoo, Bhupen Kanungo, Suresh Behera, and Sukanta Sabut. 2017. Multiresolution wavelet transform based feature extraction and ECG classification to detect cardiac abnormalities. *Measurement* 108 (2017), 55–66. https://doi.org/10.1016/j.measurement.2017.05.022

[40] Cynthia A. Sanoski. 2009. Clinical, Economic, and Quality of Life Impact of Atrial Fibrillation. *Journal of Managed Care Pharmacy* 15, 6 Supp B (Aug. 2009), 4–9. https://doi.org/10.18553/jmcp.2009.15.s6-b.4

[41] Supreeth P. Shashikumar, Amit J. Shah, Gari D. Clifford, and Shamim Nemati. 2018. Detection of Paroxysmal Atrial Fibrillation using Attention-based Bidirectional Recurrent Neural Networks. https://doi.org/10.48550/ARXIV.1805.09133

[42] Harold Smulyan. 2019. The Computerized ECG: Friend and Foe. *The American Journal of Medicine* 132, 2 (Feb. 2019), 153–160. https://doi.org/10.1016/j.amjmed.2018.08.025

[43] Sulaiman Somani, Adam J Russak, Felix Richter, Shan Zhao, Akhil Vaid, Fayzan Chaudhry, Jessica K De Freitas, Nidhi Naik, Riccardo Miotto,

Girish N Nadkarni, Jagat Narula, Edgar Argulian, and Benjamin S Glicksberg. 2021. Deep learning and the electrocardiogram: review of the current state-of-the-art. *EP Europace* 23, 8 (02 2021), 1179–1191. https://doi.org/10.1093/europace/euaa377 arXiv:https://academic.oup.com/europace/article-pdf/23/8/1179/39606181/euaa377.pdf

[44] The Arrhythmia Laboratory The Beth Israel Deaconess Medical Center. 1990. The MIT-BIH Normal Sinus Rhythm Database. https://doi.org/10.13026/C2NK5R

[45] Eric Ke Wang, liu Xi, Ruipei Sun, Fan Wang, Leyun Pan, Caixia Cheng, Antonia Dimitrakopoulou-Srauss, Nie Zhe, and Yueping Li. 2019. A new deep learning model for assisted diagnosis on electrocardiogram. *Mathematical Biosciences and Engineering* 16, 4 (2019), 2481–2491. https://doi.org/10.3934/mbe.2019124

[46] Jarosław Wasilewski and Lech Poloński. 2012. *An Introduction to ECG Interpretation*. Springer London, London, 1–20. https://doi.org/10.1007/978-0-85729-868-3_1

[47] Philip A Wolf, Janet B Mitchell, Colin S Baker, William B Kannel, and Ralph B D'Agostino. 1998. Impact of atrial fibrillation on mortality, stroke, and medical costs. *Archives of internal medicine* 158, 3 (1998), 229–234.

[48] Weiwei Wu and Hossam Haick. 2018. Materials and Wearable Devices for Autonomous Monitoring of Physiological Markers. *Advanced Materials* 30, 41 (March 2018), 1705024. https://doi.org/10.1002/adma.201705024

[49] Yong Xia, Naren Wulan, Kuanquan Wang, and Henggui Zhang. 2018. Detecting atrial fibrillation by deep convolutional neural networks. *Computers in Biology and Medicine* 93 (2018), 84–92. https://doi.org/10.1016/j.compbiomed.2017.12.007

[50] Weiyi Yang, Yujuan Si, Di Wang, and Gong Zhang. 2019. A Novel Approach for Multi-Lead ECG Classification Using DL-CCANet and TL-CCANet. *Sensors* 19, 14 (2019). https://doi.org/10.3390/s19143214

[51] Ozal Yildirim, Ulas Baran Baloglu, Ru-San Tan, Edward J. Ciaccio, and U. Rajendra Acharya. 2019. A new approach for arrhythmia classification using deep coded features and LSTM networks. *Computer Methods and Programs in Biomedicine* 176 (2019), 121–133. https://doi.org/10.1016/j.cmpb.2019.05.004

[52] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation* 31, 7 (July 2019), 1235–1270. https://doi.org/10.1162/neco_a_01199

[53] Xiangyu Zhang, Jianqing Li, Zhipeng Cai, Li Zhang, Zhenghua Chen, and Chengyu Liu. 2021. Over-fitting suppression training strategies for deep learning-based atrial fibrillation detection. *Medical &amp Biological Engineering &amp Computing* 59, 1 (Jan. 2021), 165–173. https://doi.org/10.1007/s11517-020-02292-9

A APPENDIX

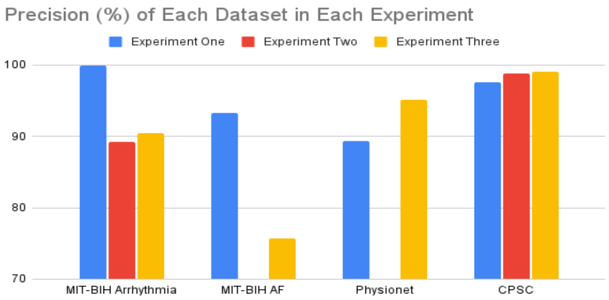

Figure 9: Bar Chart Comparing Precision of Each Dataset Used in Each Experiment

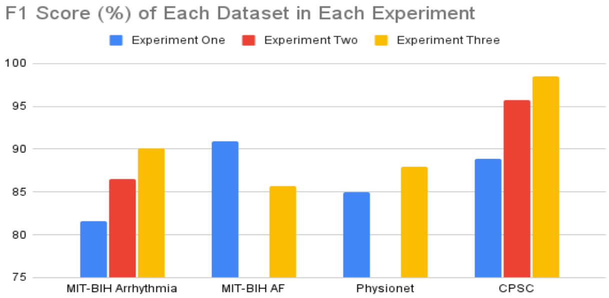

Figure 10: Bar Chart Comparing F1 Score of Each Dataset Used in Each Experiment