Atrial Fibrillation Detection from ECG Signals

Fedir Zhurba

15th December 2024

Abstract

This report presents a study on atrial fibrillation (AF) detection using a deep learning-based approach applied to raw ECG (electrocardiogram) data. The problem is motivated by the need to improve early AF detection, reduce misclassification, and enhance clinical decision-making. We experiment with a dataset of Holter-based ECG recordings and conduct exploratory data analysis (EDA), model selection, and evaluation. Our approach involves comparing various neural network architectures (including CNN, RNN, LSTM, ArNet2, and advanced CNN models) and adjusting thresholds to optimize clinical utility. We provide insights into literature, present our experimental setup and results, and conclude with future research directions.

Introduction

Background and Motivation

Atrial fibrillation is the most common form of cardiac arrhythmia, associated with an increased risk of stroke, heart failure, and mortality. Accurate and efficient detection of AF episodes can be challenging due to signal variability, noise, and subtle morphological differences in ECG waveforms. Machine learning and, more recently, deep learning methods have shown promise in automating AF detection.

Literature Review

Existing Approaches and Solutions:

- Early work used hand-crafted HRV (Heart Rate Variability) features combined with classical machine learning algorithms like SVM and gradient boosting.

- Recent research focuses on deep learning models (e.g., convolutional, recurrent, or hybrid architectures) that learn features directly from raw ECG signals.
- State-of-the-art solutions employ end-to-end deep learning frameworks and attention mechanisms, as found in top-tier conferences and journals.

Key References:

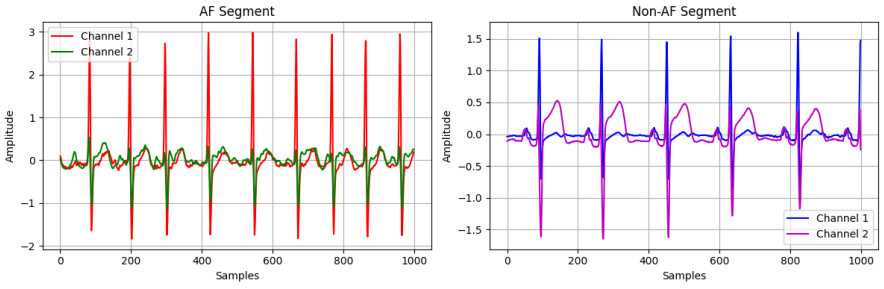
- *Top Cited Articles:*
  - Li et al. (2018) on AF detection using deep CNNs for ECG classification.
  - Rajpurkar et al. (2017) on a deep learning approach for arrhythmia detection.
- *Newest Articles:*
  - Attia, Z. I. et al. (2024) applying transformer-based models for AF detection.
  - Alamatsaz N. et al. (2024) providing hybrid CNN-LSTM explainable model for arrhythmia detection.

Methods

Data Description and Experiment Setup

Dataset:

- Holter ECG recordings with beat-level annotations.
- 100 labeled ECG recordings of Japanese population.
- Data segmentation into fixed-length windows (e.g., 5s segments at 100 Hz).
- Labels assigned as AF or non-AF based on annotation intervals.



- Distribution of the segments example:
  - Number of segments: 17220
  - Labels distribution: [AF: 7509, Non AF: 9711]

Tasks and Project Goals

Task: AF detection—classifying each ECG segment as AF or Non-AF.

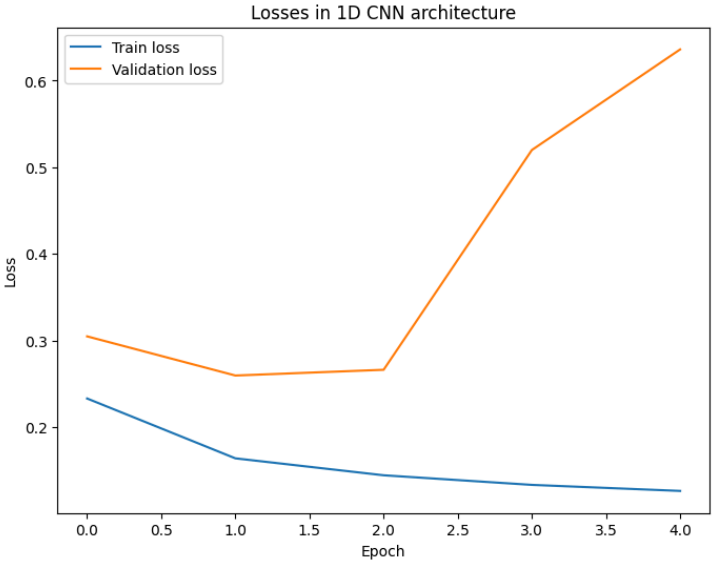
Chosen Approach and Justification:

- Deep neural networks can learn complex patterns from raw ECG signals.
- Residual connections and advanced architectures improve gradient flow and feature extraction.
- Modifying decision thresholds post-hoc can align model performance with clinical priorities.
- Estimation of the precision, recall, and f1-score to recieve

Experiment Details

Models Tested:

- Baseline CNN and RNN models.
    - Baseline CNN:
        - Losses:



        - Results:

```
Test Loss: 0.41985601309271164
              precision    recall  f1-score   support

      Non-AF       0.98      0.82      0.89    218937
          AF       0.48      0.91      0.63     40083

    accuracy                           0.83    259020
   macro avg       0.73      0.87      0.76    259020
weighted avg       0.90      0.83      0.85    259020
```

        We can see obvious overfitting due to dataset structure.
    - RNN:

- Losses:

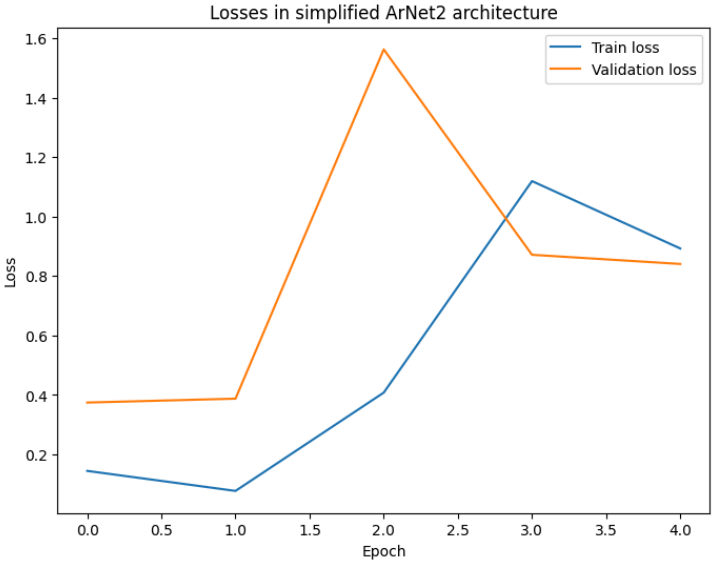
```
 20%|██        | 1/5 [10:13<40:54, 613.64s/it]
Epoch 1/5: Train Loss=9.0079, Val Loss=0.6975
 40%|████      | 2/5 [20:25<30:38, 612.77s/it]
Epoch 2/5: Train Loss=nan, Val Loss=nan
 60%|██████    | 3/5 [30:32<20:20, 610.20s/it]
Epoch 3/5: Train Loss=nan, Val Loss=nan
 80%|████████  | 4/5 [40:42<10:10, 610.06s/it]
Epoch 4/5: Train Loss=nan, Val Loss=nan
100%|██████████| 5/5 [50:54<00:00, 610.88s/it]
Epoch 5/5: Train Loss=nan, Val Loss=nan
```

- Results:

	precision	recall	f1-score	support
Non-AF	0.85	1.00	0.92	218937
AF	0.00	0.00	0.00	40083
accuracy			0.85	259020
macro avg	0.42	0.50	0.46	259020
weighted avg	0.71	0.85	0.77	259020

Failed to converge, predicted all as Non-AF.

- LSTM-based + ArNet2 model for temporal pattern exploitation.
  - Losses:

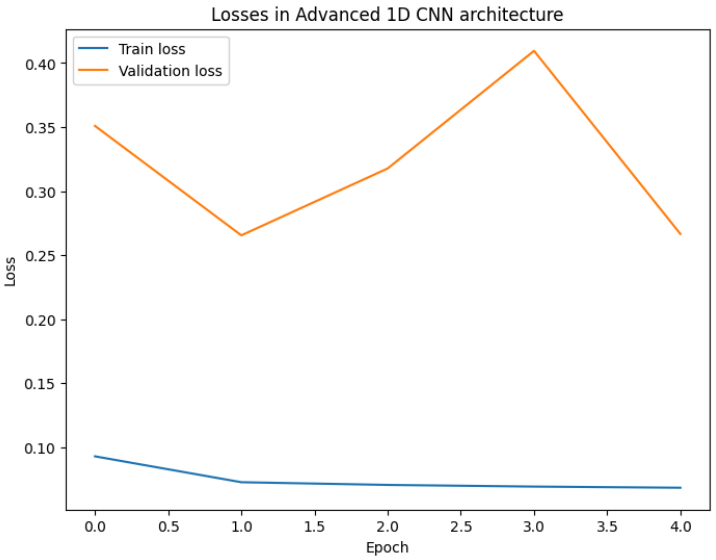


  - Results:

```
Test Loss: 1.2216176706338404
```

	precision	recall	f1-score	support
Non-AF	0.88	0.78	0.83	218937
AF	0.26	0.41	0.31	40083
accuracy			0.73	259020
macro avg	0.57	0.60	0.57	259020
weighted avg	0.78	0.73	0.75	259020

- Advanced 1D CNN: With multiple residual blocks, dropout, and global average pooling.
  - Losses:



  - Results:

```
Test Loss: 0.1939716433744944
              precision    recall  f1-score   support

      Non-AF       0.94      0.98      0.96    218937
          AF       0.87      0.64      0.74     40083

    accuracy                           0.93    259020
   macro avg       0.90      0.81      0.85    259020
weighted avg       0.93      0.93      0.93    259020
```

Advanced CNN architecture: The advanced CNN architecture builds upon the following:

- **Block1 (Conv-BN-ReLU-Conv-BN-ReLU-Pool):**
  Starts with a single input channel (1) and expands to 32 feature maps. Two convolutional layers (kernel_size=7, padding=3) extract local patterns. Batch normalization (BN) ensures stable training, and ReLU provides non-linearity. MaxPool reduces the temporal resolution by half, retaining essential features while discarding redundant details.
- **Block2 (Conv-BN-ReLU-Conv-BN-ReLU-Pool):**
  Further processes the intermediate representations, expanding from 32 to 64 channels. Another set of convolutions refines feature representations, and pooling again halves the sequence length. The network thus progressively abstracts from raw signals to higher-level patterns.

- **Residual Connection:**
  After Block2, a residual connection is introduced from the Block1 output. This 1x1 convolution on the downsampled output aligns dimensions and allows signals to bypass intermediate transformations if beneficial. Residual connections improve gradient flow, making it easier to train deeper models and reduce overfitting.
- **Block3 (Conv-BN-ReLU):**
  A third convolutional block expands channels from 64 to 128 without additional pooling. Instead of another pooling operation, global average pooling (GAP) is applied at the end to summarize the spatial dimension, resulting in a 128-dimensional feature vector.
- **Global Average Pooling & Dropout:**
  GAP replaces large fully connected layers with a simple average, reducing the parameter count and mitigating overfitting. Dropout further regularizes the final feature vector by randomly setting activations to zero during training, encouraging the model to develop robust distributed representations.
- **Fully Connected Layer:**
  The final fully connected layer maps the 128-dimensional vector to the output classes (2: AF vs Non-AF).

Threshold Adjustment:

After training our model, the default classification decision typically uses a 0.5 probability threshold. That is, if the predicted AF probability ≥ 0.5, the segment is classified as AF; otherwise, it is classified as Non-AF.

While this default threshold often maximizes balanced accuracy for many models, clinical priorities may differ. In medical diagnostics, the cost of a false positive (incorrectly classifying a Non-AF segment as AF) and the cost of a false negative (missing a genuine AF episode) are not equal. Therefore, we adjust the threshold to better align model outputs with clinical goals.

- Increasing the Threshold (e.g., from 0.5 to 0.9):
  - Pros:
    - By raising the threshold, the model becomes more conservative in labeling AF. This reduces the number of false alarms, ensuring that when the model flags AF, it is more likely correct. From a clinical perspective, fewer false positives mean less unnecessary patient anxiety, fewer

superfluous follow-up tests, and more efficient use of clinical resources.
        - If the clinical environment demands high precision (e.g., a diagnostic tool that a cardiologist uses to confirm AF), having fewer false positives enhances physician trust and can streamline decision-making.
    - Cons:
        - Increasing the threshold may cause the model to miss some genuine AF episodes (false negatives), as it now requires a stronger signal of AF before making a positive classification. This could be detrimental if the clinical scenario prioritizes early detection, as missing true AF instances might delay diagnosis or necessary treatment interventions.

```
Default Threshold (0.5) Classification Report:
              precision    recall  f1-score   support

      Non-AF       0.94      0.98      0.96    218937
          AF       0.87      0.64      0.74     40083

    accuracy                           0.93    259020
   macro avg       0.90      0.81      0.85    259020
weighted avg       0.93      0.93      0.93    259020


Higher Threshold (0.9) Classification Report:
              precision    recall  f1-score   support

      Non-AF       0.89      1.00      0.94    218937
          AF       0.95      0.35      0.51     40083

    accuracy                           0.90    259020
   macro avg       0.92      0.67      0.73    259020
weighted avg       0.90      0.90      0.88    259020
```

- Decreasing the Threshold (e.g., from 0.5 down to 0.1 and 0.02):
    - Pros:
        - Lowering the threshold makes the model more sensitive to any indication of AF, reducing false negatives. Clinically, this is vital if the primary goal is to catch every possible AF episode, especially in a screening context or in high-risk populations where missing an AF event could lead to significant morbidity. It ensures the model is less likely to overlook early or subtle signs of AF.
    - Cons:
        - A lower threshold may increase the false positive rate, labeling more Non-AF segments as AF. While this ensures fewer missed AF cases, it also leads to more false alarms,

potentially causing patient anxiety, unnecessary follow-up tests, and additional workload for healthcare providers. In a setting where resource constraints and patient experience are critical, a higher false positive rate could be problematic.

```
Lower Threshold (0.1) Classification Report:
              precision    recall  f1-score   support

      Non-AF       0.97      0.92      0.94    218937
          AF       0.66      0.83      0.74     40083

    accuracy                           0.91    259020
   macro avg       0.82      0.88      0.84    259020
weighted avg       0.92      0.91      0.91    259020


Lowest Threshold (0.02) Classification Report:
              precision    recall  f1-score   support

      Non-AF       0.99      0.81      0.89    218937
          AF       0.48      0.96      0.64     40083

    accuracy                           0.83    259020
   macro avg       0.74      0.88      0.77    259020
weighted avg       0.91      0.83      0.85    259020
```

Results

Model Performance:

- Baseline CNN architecture shows fine performance, but is very sensitive to overfitting, requires more advanced techiques.
- ArNet2 and LSTM perform well on training, but require more advanced architecture and more training time to perform well.
- RNNs fails to converge, and makes very overfitted prediction, needs much more advanced architecture to avoid single-class predictions.
- Enhanced CNN models achieve best results due to high-level architecure.
- After adjusting the threshold, precision improves at the expense of recall, showing the trade-off between false alarms and missed AF events.

Discussion

Interpretation of Results:

- Residual connections and deeper architectures improve representation learning and reduce overfitting.
- Threshold adjustments allow clinicians to tune the model to minimize false positives (increasing precision) or minimize false negatives (improving sensitivity), depending on the clinical scenario.

Conclusions and Future Work

- The advanced CNN architectures outperformed simpler models in detecting AF from ECG segments.
- Threshold tuning is crucial in tailoring the model's output to specific clinical needs.
- Future research could explore:
    - Transformer-based architectures or attention mechanisms for improved sequence modeling.
    - Domain adaptation techniques for generalizing across patient populations.
    - Integration with other physiological data for multi-modal arrhythmia detection.

References

1. Acharya, U. R. et al. (2018). "A deep convolutional neural network model to classify heartbeats".
2. Hannun, A. Y. et al. (2019). "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network".
3. Attia, Z. I. et al. (2024). "Artificial intelligence-enhanced electrocardiography for accurate diagnosis and management of cardiovascular diseases".
4. Alamatsaz N. et al. (2024). "A lightweight hybrid CNN-LSTM explainable model for ECG-based arrhythmia detection".
5. Tsutsui, K. et al. (2024). "SHDB-AF: a Japanese Holter ECG database of atrial fibrillation (version 1.0.0)". *PhysioNet*.

Code repository:

- GitHub: *https://github.com/fazhur/ecg_af_prediction*