



Atrial Fibrillation Events Detection Website

Darren Liu



Introduction

Motivation: Atrial fibrillation (AF) is the most frequent arrhythmia, but people are having a hard time detecting it in early stage.

Objectives and goals: My goal is to develop a website that can classify AF from PPG signals.



Methodology

The data I used:

The data is sourced from UCLA's clinical dataset. It's a huge dataset with more than 20GB of PPG signals recorded from different hospitals and devices.



Methodology

Tools I used:

PyTorch for building neural network

React.JS for front-end development

React-plotly.js for plotting


Flask for back-end development

AWS and Apache2 for deployment



Methodology

Front-end page:

Drag and drop your .csv file to the box below 

(Please only include no more than 10 records, each containing 30s of signals. [See this example file](#))





Methodology

Front-end page:

Drag and drop your .csv file to the box below 

(Please only include no more than 10 records, each containing 30s of signals. [See this example file](#))



test2.csv



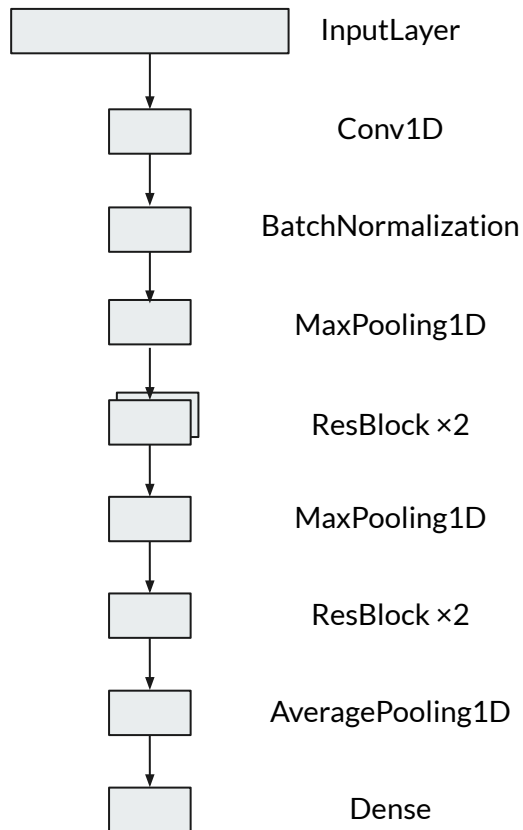
UPLOAD

CLEAR



Methodology

Back-end server:



Methodology

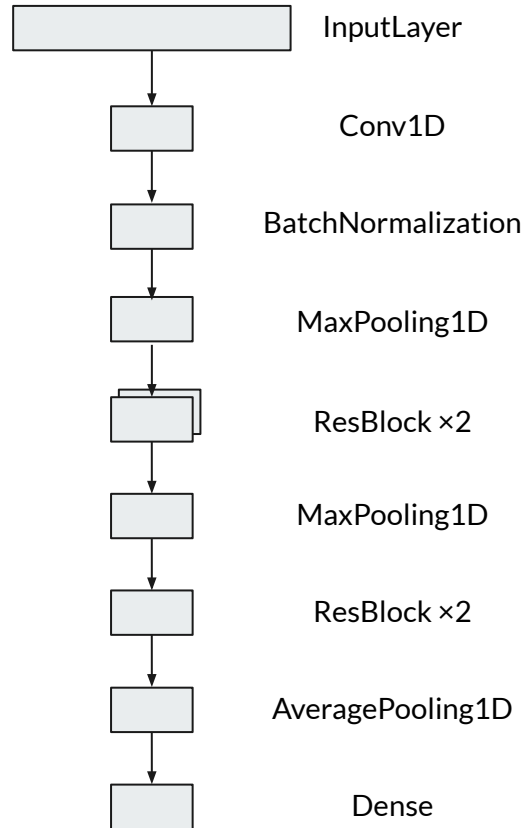
Back-end server:

{'data': [list of signals in the file],

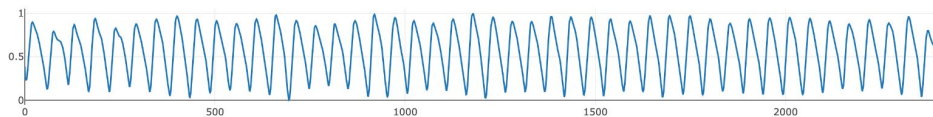
'pred': [1, 1, 0, 1, 0],

'pred_prob': [0.9, 0.3, -0.4, 0.5, -0.8],

'error': 0}



Results



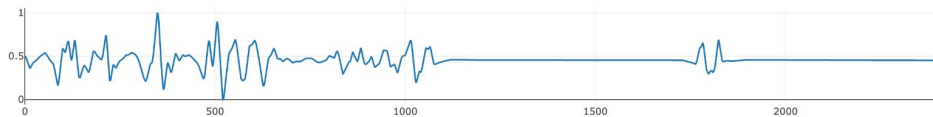
Non-AF
Confidence: 2.18

The 6th signal



AF
Confidence: 0.02

The 7th signal



Non-AF
Confidence: 0.15



Conclusions

Nowadays, more people have access to clinical data like PPG, ECG signals at home by smart devices and wearable devices, building a website that allows them to take advantage of machine learning models to detect AF in early stage and help them receive treatments in time is essential.



Future Work

1. To deploy the back-end server to one of the Emory University lab machines to make it accessible online.
2. To show more metadata on the page and how the model works to make the results and the model more clear to users.



Thanks for your attention!