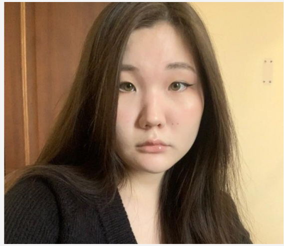
Final Presentation

Michael J Fox Foundation 1A

November 1, 2025

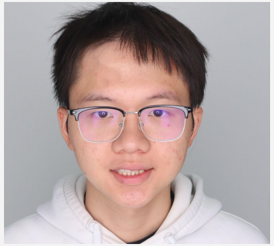
Our Team



Saniya Mukhambetkaliyeva

Bryn Mawr College ‘27

Computer Science



Jeriel Goh

San Jose State University ‘27
Computer Engineering

Bhuvana Kotha

University of California, Riverside
‘26
Data Science

Mehek Bhatnagar

University of California, Irvine ‘27.
Computer Science.

Marc Romero

University of California, Santa Cruz
‘26
Computer Engineering

Ramya Madugula

University of California, Riverside
‘26
Computer Science

Did you know that the number of people with Parkinson’s disease is projected to more than double globally by 2025!

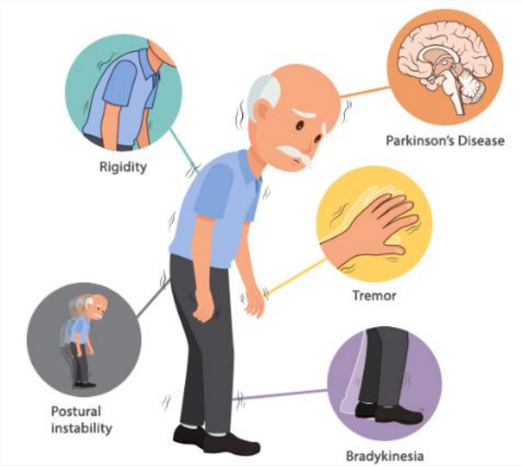
Introduction

What do you know about Parkinson's Disease?

Parkinson's disease (PD) is the fastest growing neurodegenerative disorder, affecting approximately 1% of individuals in the U.S. over the age of 60 (Willis et al. 2022).

Freezing of Gait (FoG) is a debilitating condition that affects many people with Parkinson's Disease that leads to depression, increased risk of falls, and a loss of independence

There is no clear understanding of what causes it.



Introduction

Why is Detecting FoG important?

Detecting and quantifying these episodes allows researchers to:

- Measure disease progression more precisely.
- Test how well new drugs or therapies reduce FOG events.
- Develop personalized treatments that target these movement disruptions.

Project Overview

Problem Statement

THE MICHAEL J. FOX FOUNDATION
FOR PARKINSON'S RESEARCH

OBJECTIVE
Identify the start and stop of FoG episodes by detecting 3 FoG events:

- Start Hesitation
- Turning
- Walking

DESIRED OUTCOME
Develop a machine learning model trained on data collected from a wearable 3D lower back sensor to detect freezing of gait in subjects with Parkinson's Disease.

The model should predict whether a FoG episode is occurring at a given timestamp, producing a binary classification: True or False.

Methodology & Approach

1. Understand Data with EDA + Research

2. Data Visualizations & Preprocessing

3. Feature Engineering

4. Model Training

5. Model Evaluation

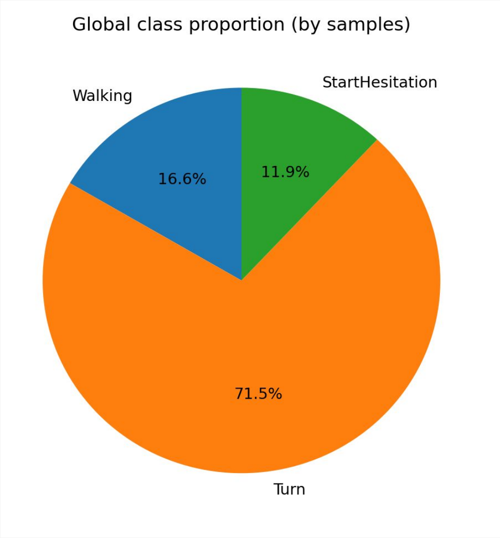
Data Understanding & Preparation

Introducing the Data

Data from the MJFF FoG Kaggle competition 2023

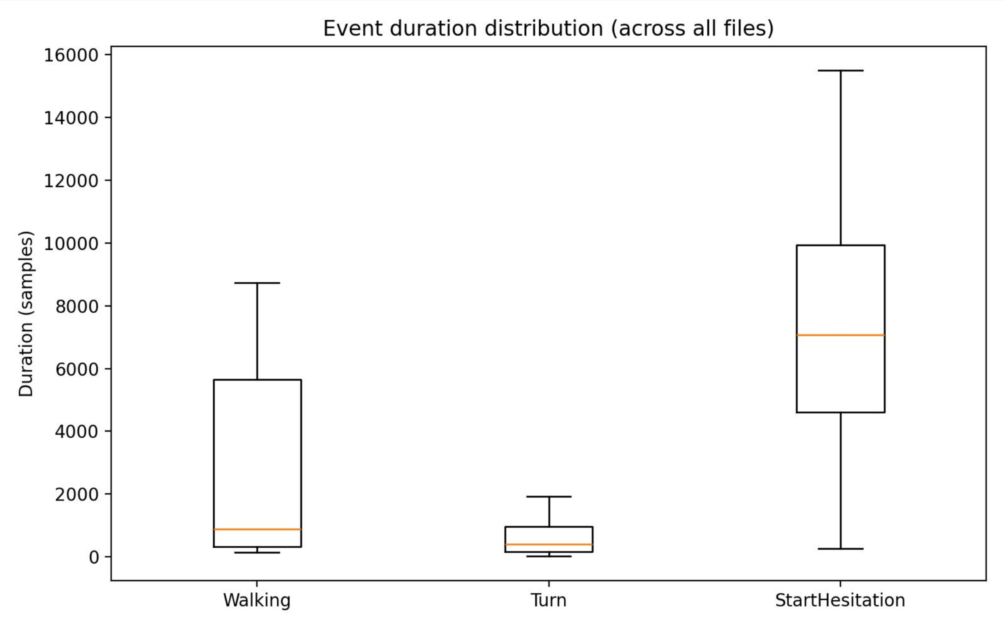
Sourced from 3D wearable accelerometers on subjects exhibiting freezing of gait episodes. Split into 3 datasets:

- tDCS FOG (tdcsfog) dataset, comprising data series collected in the lab
- DeFOG (defog) dataset, comprising data series collected in the subject's home
- The Daily Living (daily) dataset, comprising one week of continuous 24/7



Dataset Columns: Time, AccV, AccML, AccAP, and StartHesitation, Turn, Walking (labeled 0 or 1).

Data Distribution



All: 3
one: 52
Turn: 40
Turn & StartHesitation: 1
Walking & Turn: 4

Libraries Used

matplotlib

SciPy

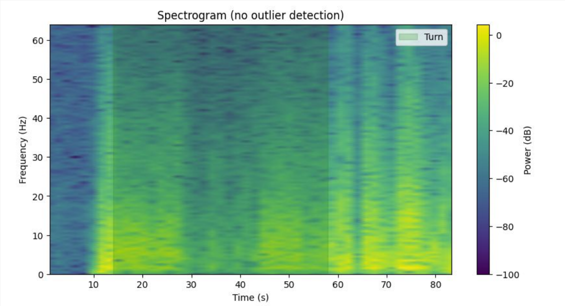
tsfresh

PyTorch

Polars

Exploring the Data

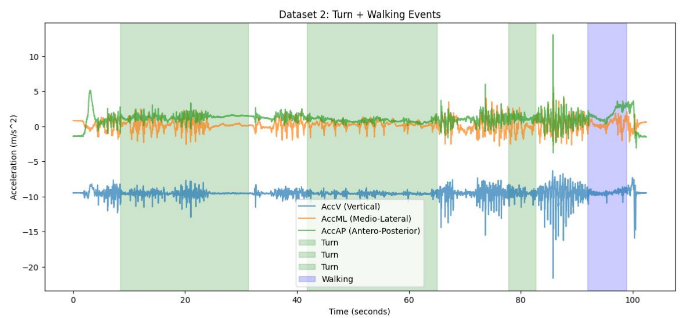
Spectrograms



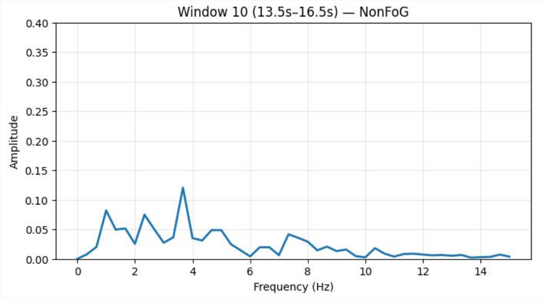
Sampling Large Subsets

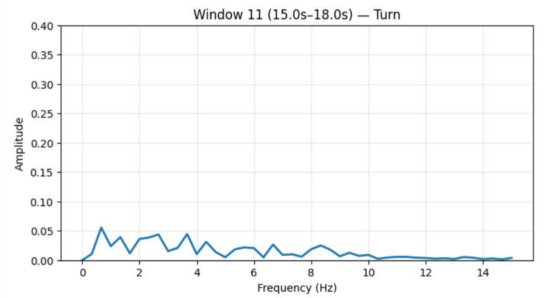
```
I am starting my quantitative analysis on the 100 datasets...
Processing Datasets:   0%|          | 0/100 [00:00<?, ?it/s]
======================================================================
My FOG Event Analysis Summary (100 Datasets)
======================================================================
--- StartHesitation ---
  Total Occurrences:           9
  Average Duration:            49.273 seconds
--- Turn ---
  Total Occurrences:           104
  Average Duration:            16.434 seconds
--- Walking ---
  Total Occurrences:           4
  Average Duration:            3.09 seconds
```

Visualizing Acc vs. Time



FFT





Features included in the Model

Base Variables

“start_time”: s / fs,
“end_time”: e / fs,
“label”: binary_label,

Magnitude Features

“low_energy”: low_mag,
“high_energy”: high_mag,
“total_energy”: total_mag,
“energy_ratio”: energy_ratio,
“freeze_index”: freeze_index,

V-axis Features (Vertical)

“V_low_energy”: V_low,
“V_high_energy”: V_high,
“V_total_energy”: V_total,

ML-axis Features (Medio-Lateral)

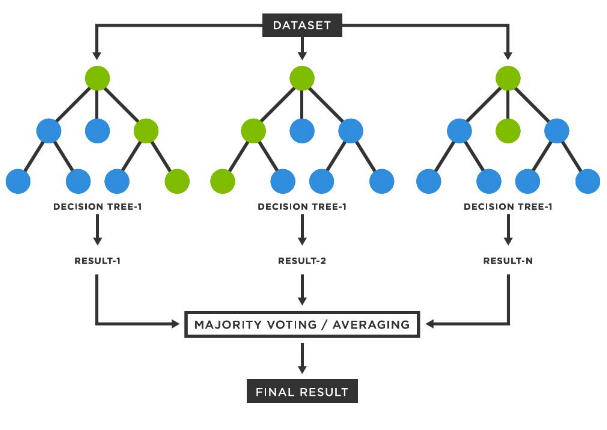
“ML_low_energy”: ML_low,
“ML_high_energy”: ML_high,
“ML_total_energy”: ML_total,

AP-axis Features (Antero-Posterior)

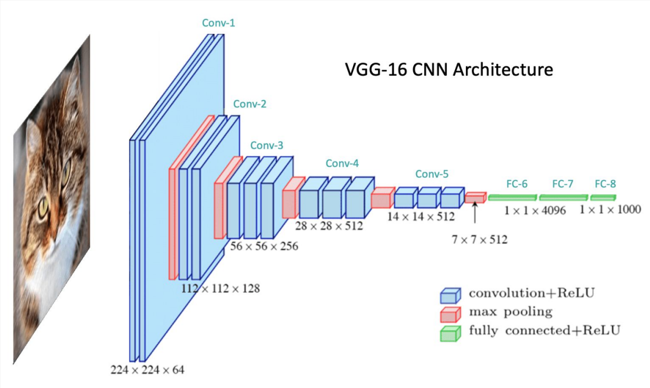
“AP_low_energy”: AP_low,
“AP_high_energy”: AP_high,
“AP_total_energy”: AP_total,

Modeling & Evaluation

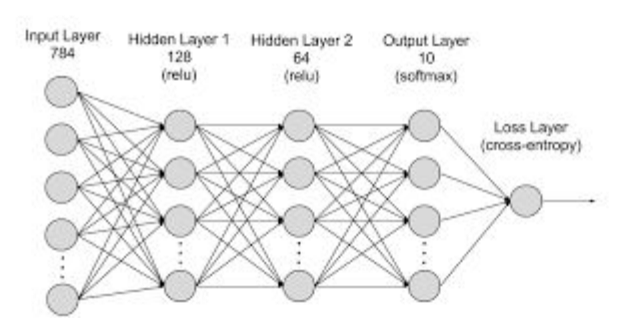
Models Considered



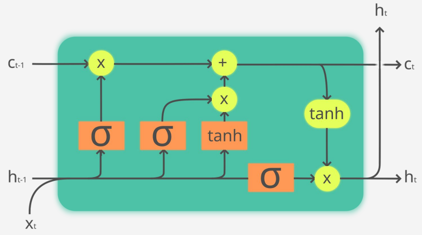
Decision Tree/XG Boost



Convolutional Neural Network (CNN)



Recurrent Neural Networks (RNN)



Long Short-Term Memory (LSTM, a Specific RNN)

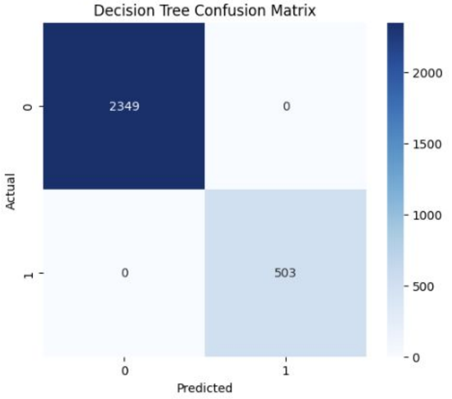
Things to Note:

```
# 1. SETUP EARLY STOPPING CONFIG
num_epochs = 20
patience = 5      # Stop if no improvement after 5 epochs
```

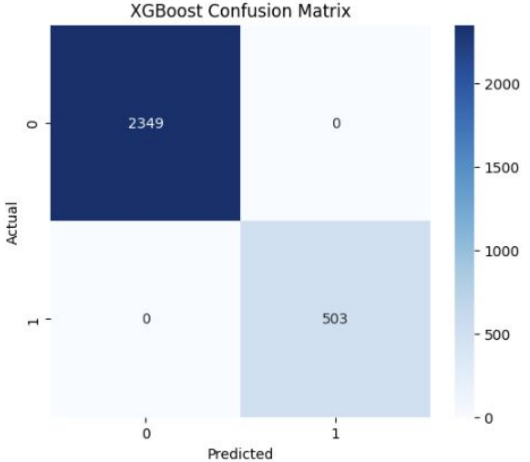
- Implemented function to stop training the model if val_loss plateaus.
- Best accuracy is highlighted by the red circle.
- Helps save time and computing power

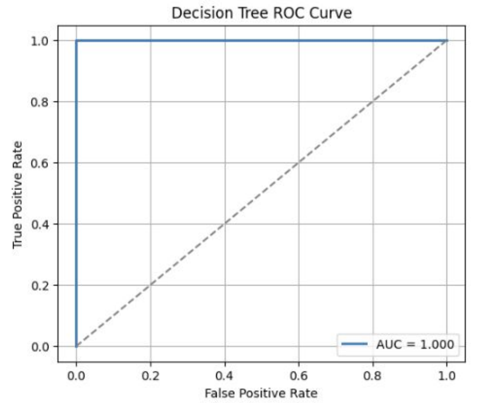
Tree Models: DT, XGBoost

```
===== Decision Tree =====
Accuracy : 1.0
Precision: 1.0
Recall   : 1.0
F1 Score : 1.0

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2349
           1       1.00      1.00      1.00       503

    accuracy                           1.00      2852
   macro avg       1.00      1.00      1.00      2852
weighted avg       1.00      1.00      1.00      2852

AUC      : 1.0
```




```
===== XGBoost =====
Accuracy : 1.0
Precision: 1.0
Recall   : 1.0
F1 Score : 1.0

Classification Report:
              precision    recall  f1-score

           0       1.00      1.00      1.00
           1       1.00      1.00      1.00

    accuracy                           1.00
   macro avg       1.00      1.00      1.00
weighted avg       1.00      1.00      1.00

AUC      : 1.0
```

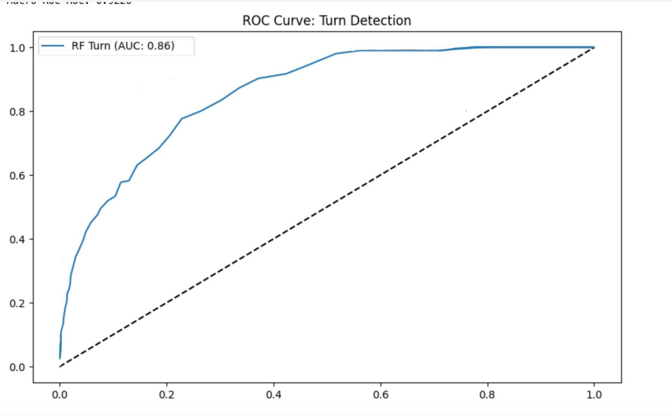




- Decision model is perfect → This raises concerns
- Models are overfitting and not able to handle noise sensitive data
- Trees make piecewise predictions, they struggle to model gradual changes in signals

Tree Models: RF

```
--- Random Forest Evaluation ---
                precision    recall  f1-score   support

        Normal       0.80      0.99      0.89       978
          Turn       0.79      0.15      0.25       206
       Walking       0.00      0.00      0.00         3
StartHesitation       0.68      0.21      0.32        73

      accuracy                           0.80      1260
     macro avg       0.57      0.34      0.36      1260
  weighted avg       0.79      0.80      0.75      1260

Macro Recall (Sensitivity): 0.3354
Macro ROC AUC: 0.8740
```



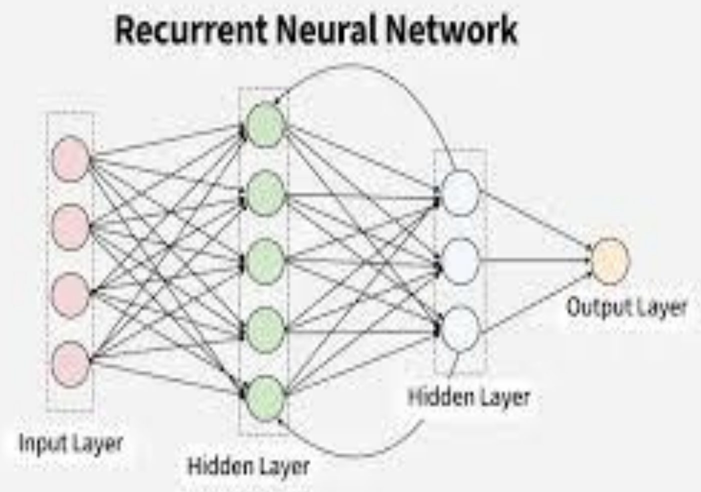
- Random Forest AUC = 0.86
- Clinical Impact
- ROC Curve →detection ability
- Overfitting Risk

How to improve the model in the future:

- Hyperparameter tune, scale positions
- Make the model pruning based
- Test the models on the testing data

RNN Models

- Input: Sliding windows of IMU gait data → sequences of shape [time steps × features]
- Temporal modeling: GRU-based RNN updates a hidden state at each timestep to capture recent movement history
- Output: For every timestep, the model predicts a FoG probability (FoG vs non-FoG)
- Training: Uses binary cross-entropy loss on per-timestep labels to match true FoG annotations
- Why RNNs?: FoG is a time-dependent event, so RNNs can pick up patterns and changes over time, not just single points



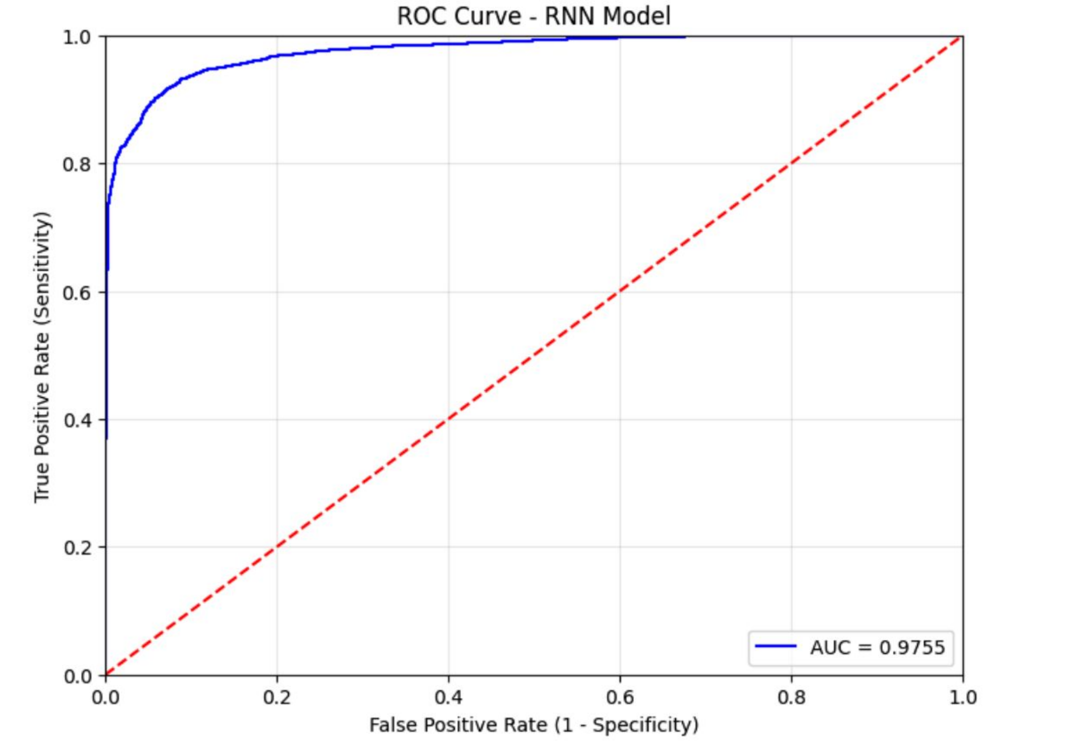
RNN Model Accuracy

```
Starting training with Early Stopping...
Epoch 1   | Train Loss: 0.5439 | Val Loss: 0.5188 | Val Acc: 0.7620 | Val Prec: 0.9370 | Val Rec: 0.4557
    --> Best model saved (Loss: 0.5188)
Epoch 2   | Train Loss: 0.4202 | Val Loss: 0.4261 | Val Acc: 0.8578 | Val Prec: 0.9098 | Val Rec: 0.7289
    --> Best model saved (Loss: 0.4261)
Epoch 3   | Train Loss: 0.3873 | Val Loss: 0.4195 | Val Acc: 0.8682 | Val Prec: 0.9502 | Val Rec: 0.7194
    --> Best model saved (Loss: 0.4195)
Epoch 4   | Train Loss: 0.3702 | Val Loss: 0.3787 | Val Acc: 0.8698 | Val Prec: 0.9271 | Val Rec: 0.7440
    --> Best model saved (Loss: 0.3787)
Epoch 5   | Train Loss: 0.3641 | Val Loss: 0.3764 | Val Acc: 0.8742 | Val Prec: 0.9515 | Val Rec: 0.7335
    --> Best model saved (Loss: 0.3764)
Epoch 6   | Train Loss: 0.3483 | Val Loss: 0.3617 | Val Acc: 0.8770 | Val Prec: 0.9530 | Val Rec: 0.7394
    --> Best model saved (Loss: 0.3617)
Epoch 7   | Train Loss: 0.3358 | Val Loss: 0.3424 | Val Acc: 0.8851 | Val Prec: 0.9469 | Val Rec: 0.7654
    --> Best model saved (Loss: 0.3424)
Epoch 8   | Train Loss: 0.3330 | Val Loss: 0.3293 | Val Acc: 0.8937 | Val Prec: 0.9335 | Val Rec: 0.8003
    --> Best model saved (Loss: 0.3293)
Epoch 9   | Train Loss: 0.3313 | Val Loss: 0.3187 | Val Acc: 0.8973 | Val Prec: 0.9255 | Val Rec: 0.8178
    --> Best model saved (Loss: 0.3187)
Epoch 10  | Train Loss: 0.3245 | Val Loss: 0.3183 | Val Acc: 0.8935 | Val Prec: 0.9623 | Val Rec: 0.7730
    --> Best model saved (Loss: 0.3183)
Epoch 11  | Train Loss: 0.3149 | Val Loss: 0.3181 | Val Acc: 0.8923 | Val Prec: 0.9689 | Val Rec: 0.7645
    --> Best model saved (Loss: 0.3181)
Epoch 12  | Train Loss: 0.3068 | Val Loss: 0.3213 | Val Acc: 0.8897 | Val Prec: 0.9823 | Val Rec: 0.7470
    --> No improvement. Patience: 1/5
Epoch 13  | Train Loss: 0.2977 | Val Loss: 0.2939 | Val Acc: 0.9019 | Val Prec: 0.9632 | Val Rec: 0.7933
    --> Best model saved (Loss: 0.2939)
Epoch 14  | Train Loss: 0.2880 | Val Loss: 0.2914 | Val Acc: 0.8993 | Val Prec: 0.9650 | Val Rec: 0.7853
    --> Best model saved (Loss: 0.2914)
Epoch 15  | Train Loss: 0.2815 | Val Loss: 0.3030 | Val Acc: 0.8937 | Val Prec: 0.9783 | Val Rec: 0.7602
    --> No improvement. Patience: 1/5
Epoch 16  | Train Loss: 0.2741 | Val Loss: 0.2816 | Val Acc: 0.9058 | Val Prec: 0.9691 | Val Rec: 0.7979
    --> Best model saved (Loss: 0.2816)
Epoch 17  | Train Loss: 0.2583 | Val Loss: 0.2583 | Val Acc: 0.9167 | Val Prec: 0.9553 | Val Rec: 0.8380
    --> Best model saved (Loss: 0.2583)
Epoch 18  | Train Loss: 0.2470 | Val Loss: 0.2530 | Val Acc: 0.9182 | Val Prec: 0.9426 | Val Rec: 0.8545
    --> Best model saved (Loss: 0.2530)
Epoch 19  | Train Loss: 0.2494 | Val Loss: 0.2843 | Val Acc: 0.9090 | Val Prec: 0.9775 | Val Rec: 0.7985
    --> No improvement. Patience: 1/5
Epoch 20  | Train Loss: 0.2457 | Val Loss: 0.2437 | Val Acc: 0.9248 | Val Prec: 0.9239 | Val Rec: 0.8919
    --> Best model saved (Loss: 0.2437)
Training Complete. Best Validation Loss: 0.2437
```

Training Loss and Validation Loss decreases, meaning no overfitting

- Val Accuracy = 0.9248 → on unseen data, the model correctly labels about 92–93% of timesteps as FoG vs non-FoG.
- That means out of 100 labeled timesteps, roughly 93 are classified correctly.
- Val Precision = 0.9239 → when the model predicts FoG, it's right about 92% of the time (few false alarms).
- Val Recall = 0.8919 → it catches about 89% of the true FoG timesteps (misses some, but most are detected).
- These metrics are computed on the validation set, so they reflect how well the RNN generalizes to new gait sequences, not just the training data.

RNN Model AUC-ROC



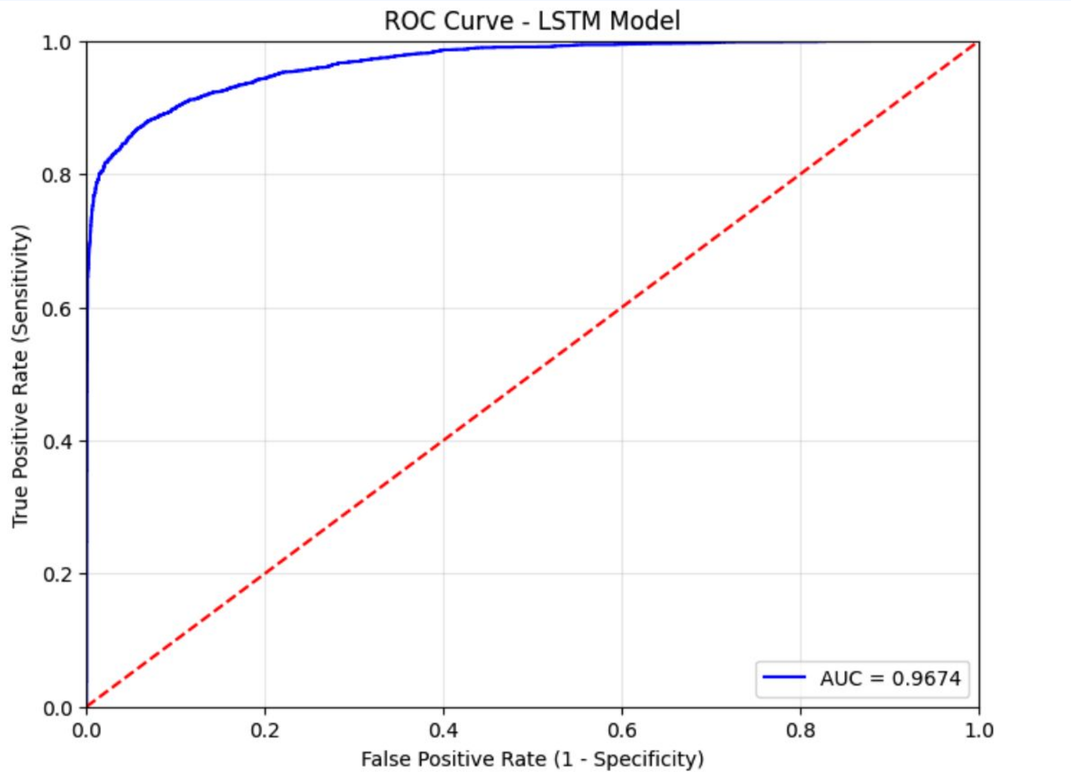
- This is the ROC curve for our RNN FoG model: it shows the trade-off between True Positive Rate on the y-axis and False Positive Rate on the x-axis.
- The red dashed diagonal is a random classifier (no skill).
- Our blue ROC curve hugs the top-left corner, meaning we get high sensitivity even at low false positive rates.
- The AUC = 0.9755 means the model can correctly rank FoG vs non-FoG timesteps about 98% of the time, showing very strong discriminative performance on the validation data.

LSTM Models – a specific RNN

```
Epoch    Train Loss    Val Loss     Val Acc
----------------------------------------------------------------
Epoch 0  : Train Loss: 0.5058 | Val Loss: 0.4934 | Val Acc: 0.8232 | Val Prec: 0.8826 | Val Rec: 0.6609
    --> New best model saved! (Val Loss: 0.4934)
Epoch 1  : Train Loss: 0.4123 | Val Loss: 0.4870 | Val Acc: 0.8311 | Val Prec: 0.9104 | Val Rec: 0.6567
    --> New best model saved! (Val Loss: 0.4870)
Epoch 2  : Train Loss: 0.3864 | Val Loss: 0.4199 | Val Acc: 0.8421 | Val Prec: 0.9327 | Val Rec: 0.6668
    --> New best model saved! (Val Loss: 0.4199)
Epoch 3  : Train Loss: 0.3611 | Val Loss: 0.3840 | Val Acc: 0.8830 | Val Prec: 0.9343 | Val Rec: 0.7715
    --> New best model saved! (Val Loss: 0.3840)
Epoch 4  : Train Loss: 0.3398 | Val Loss: 0.3501 | Val Acc: 0.8865 | Val Prec: 0.9465 | Val Rec: 0.7694
    --> New best model saved! (Val Loss: 0.3501)
Epoch 5  : Train Loss: 0.3410 | Val Loss: 0.4035 | Val Acc: 0.8300 | Val Prec: 0.9711 | Val Rec: 0.6074
    --> No improvement. Patience: 1/5
Epoch 6  : Train Loss: 0.3265 | Val Loss: 0.3479 | Val Acc: 0.8807 | Val Prec: 0.8551 | Val Rec: 0.8570
    --> New best model saved! (Val Loss: 0.3479)
Epoch 7  : Train Loss: 0.3349 | Val Loss: 0.3496 | Val Acc: 0.8846 | Val Prec: 0.9674 | Val Rec: 0.7464
    --> No improvement. Patience: 1/5
Epoch 8  : Train Loss: 0.3081 | Val Loss: 0.3270 | Val Acc: 0.8954 | Val Prec: 0.9257 | Val Rec: 0.8126
    --> New best model saved! (Val Loss: 0.3270)
Epoch 9  : Train Loss: 0.3094 | Val Loss: 0.3166 | Val Acc: 0.8969 | Val Prec: 0.9683 | Val Rec: 0.7764
    --> New best model saved! (Val Loss: 0.3166)
Epoch 10 : Train Loss: 0.2960 | Val Loss: 0.3254 | Val Acc: 0.8949 | Val Prec: 0.9699 | Val Rec: 0.7700
    --> No improvement. Patience: 1/5
Epoch 11 : Train Loss: 0.2866 | Val Loss: 0.2934 | Val Acc: 0.9060 | Val Prec: 0.9463 | Val Rec: 0.8196
    --> New best model saved! (Val Loss: 0.2934)
Epoch 12 : Train Loss: 0.3033 | Val Loss: 0.3108 | Val Acc: 0.8963 | Val Prec: 0.9310 | Val Rec: 0.8095
    --> No improvement. Patience: 1/5
Epoch 13 : Train Loss: 0.2839 | Val Loss: 0.3312 | Val Acc: 0.8875 | Val Prec: 0.9767 | Val Rec: 0.7461
    --> No improvement. Patience: 2/5
Epoch 14 : Train Loss: 0.2778 | Val Loss: 0.3078 | Val Acc: 0.9034 | Val Prec: 0.9732 | Val Rec: 0.7884
    --> No improvement. Patience: 3/5
Epoch 15 : Train Loss: 0.2675 | Val Loss: 0.2787 | Val Acc: 0.9105 | Val Prec: 0.9227 | Val Rec: 0.8554
    --> New best model saved! (Val Loss: 0.2787)
Epoch 16 : Train Loss: 0.2682 | Val Loss: 0.2699 | Val Acc: 0.9144 | Val Prec: 0.9478 | Val Rec: 0.8395
    --> New best model saved! (Val Loss: 0.2699)
Epoch 17 : Train Loss: 0.2643 | Val Loss: 0.2688 | Val Acc: 0.9098 | Val Prec: 0.9146 | Val Rec: 0.8628
    --> New best model saved! (Val Loss: 0.2688)
Epoch 18 : Train Loss: [illegible] | Val Loss: 0.2763 | Val Acc: 0.9102 | Val Prec: [illegible] | Val Rec: 0.8101
    --> No improvement. Patience: 1/5
Epoch 19 : Train Loss: 0.2598 | Val Loss: 0.2638 | Val Acc: 0.9120 | Val Prec: 0.9222 | Val Rec: 0.8600
    --> New best model saved! (Val Loss: 0.2638)

Training finished. Loaded best model with Val Loss: 0.2638
```

- LSTM (Long Short Term Memory) Models are a more advanced version of RNN, excelling in capturing patterns of longer sequences
- Validation Accuracy: 0.9128 (0.9248 in RNN)
- Validation Precision: 0.9222 (0.9239 in RNN)
- Validation Recall: 0.8600 (0.8919 in RNN)
- Similar results to RNN
  - Sequences are short enough to be comprehended by a simple RNN

LSTM Model AUC-ROC

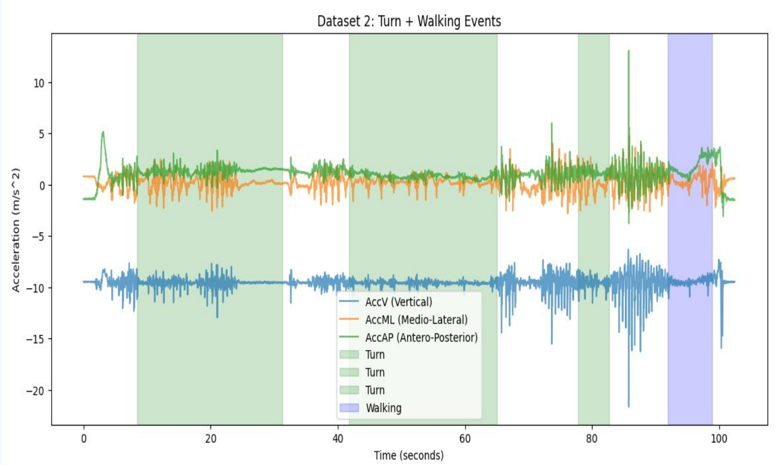


- LSTM (Long Short Term Memory) Models are a more advanced version of RNN, excelling in capturing patterns of longer sequences
- Validation Accuracy: 0.9128 (0.9248 in RNN)
- Validation Precision: 0.9222 (0.9239 in RNN)
- Validation Recall: 0.8600 (0.8919 in RNN)
- Similar results to RNN
  - Sequences are short enough to be comprehended by a simple RNN

CNN Model Accuracy

```
Epoch     Train Loss    Val Loss      Val Acc
------------------------------------------------------------------
Epoch 0  : Train Loss: 0.6417 | Val Loss: 0.5148 | Val Acc: 0.8036 | Val Prec: 0.8969 | Val Rec: 0.5939
    --> New best model saved! (Val Loss: 0.5148)
Epoch 1  : Train Loss: 0.4420 | Val Loss: 0.4450 | Val Acc: 0.8430 | Val Prec: 0.8608 | Val Rec: 0.7406
    --> New best model saved! (Val Loss: 0.4450)
Epoch 2  : Train Loss: 0.4002 | Val Loss: 0.4154 | Val Acc: 0.8523 | Val Prec: 0.8758 | Val Rec: 0.7495
    --> New best model saved! (Val Loss: 0.4154)
Epoch 3  : Train Loss: 0.3883 | Val Loss: 0.3952 | Val Acc: 0.8591 | Val Prec: 0.8306 | Val Rec: 0.8288
    --> New best model saved! (Val Loss: 0.3952)
Epoch 4  : Train Loss: 0.3767 | Val Loss: 0.3838 | Val Acc: 0.8641 | Val Prec: 0.8547 | Val Rec: 0.8092
    --> New best model saved! (Val Loss: 0.3838)
Epoch 5  : Train Loss: 0.3575 | Val Loss: 0.3991 | Val Acc: 0.8395 | Val Prec: 0.9348 | Val Rec: 0.6582
    --> No improvement. Patience: 1/5
Epoch 6  : Train Loss: 0.3562 | Val Loss: 0.3792 | Val Acc: 0.8580 | Val Prec: 0.9168 | Val Rec: 0.7225
    --> New best model saved! (Val Loss: 0.3792)
Epoch 7  : Train Loss: 0.3668 | Val Loss: 0.4087 | Val Acc: 0.8373 | Val Prec: 0.9525 | Val Rec: 0.6389
    --> No improvement. Patience: 1/5
Epoch 8  : Train Loss: 0.3403 | Val Loss: 0.4058 | Val Acc: 0.8288 | Val Prec: 0.9549 | Val Rec: 0.6156
    --> No improvement. Patience: 2/5
Epoch 9  : Train Loss: 0.3377 | Val Loss: 0.3781 | Val Acc: 0.8486 | Val Prec: 0.9416 | Val Rec: 0.6763
    --> New best model saved! (Val Loss: 0.3781)
Epoch 10 : Train Loss: 0.3245 | Val Loss: 0.3892 | Val Acc: 0.8392 | Val Prec: 0.9508 | Val Rec: 0.6450
    --> No improvement. Patience: 1/5
Epoch 11 : Train Loss: 0.3439 | Val Loss: 0.3578 | Val Acc: 0.8581 | Val Prec: 0.9282 | Val Rec: 0.7124
    --> New best model saved! (Val Loss: 0.3578)
Epoch 12 : Train Loss: 0.3356 | Val Loss: 0.3506 | Val Acc: 0.8714 | Val Prec: 0.9021 | Val Rec: 0.7734
    --> New best model saved! (Val Loss: 0.3506)
Epoch 13 : Train Loss: 0.3293 | Val Loss: 0.3490 | Val Acc: 0.8724 | Val Prec: 0.9079 | Val Rec: 0.7700
    --> New best model saved! (Val Loss: 0.3490)
Epoch 14 : Train Loss: 0.3137 | Val Loss: 0.3603 | Val Acc: 0.8462 | Val Prec: 0.9304 | Val Rec: 0.6631
    --> No improvement. Patience: 1/5
Epoch 15 : Train Loss: 0.3216 | Val Loss: 0.3764 | Val Acc: 0.8501 | Val Prec: 0.9586 | Val Rec: 0.6668
    --> No improvement. Patience: 2/5
Epoch 16 : Train Loss: 0.3091 | Val Loss: 0.3544 | Val Acc: 0.8563 | Val Prec: 0.9457 | Val Rec: 0.6928
    --> No improvement. Patience: 3/5
Epoch 17 : Train Loss: 0.3092 | Val Loss: 0.3953 | Val Acc: 0.8367 | Val Prec: 0.9727 | Val Rec: 0.6230
    --> No improvement. Patience: 4/5
Epoch 18 : Train Loss: 0.3129 | Val Loss: 0.3809 | Val Acc: 0.8449 | Val Prec: 0.9705 | Val Rec: 0.6450
    --> No improvement. Patience: 5/5
Early stopping triggered!

Training finished. Loaded best model with Val Loss: 0.3490
```

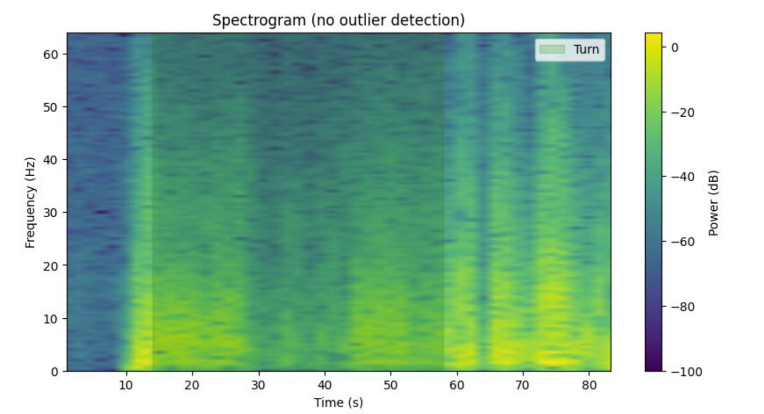
- CNN recognizes the 2D image of signal data and makes a prediction
- Reason for similar accuracy is because we fed CNN with raw signal data instead of a 2D image. We learned that it is important to understand a model first before using it.



Final Thoughts

Roadblocks & Future Developments

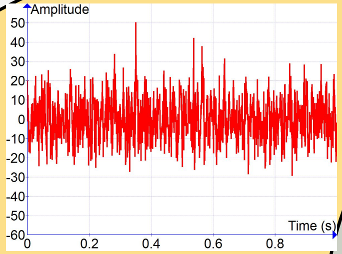
- Reevaluate modelling process.. Accuracy of model too high, mistakes with preprocessing steps
- Explore deep learning models using spectrogram data.
- Investigate the use of Root Mean Square to measure motion intensity as a new feature
- Improve our model to identify specific cases of FOG events based on a FOG event instead of just a binary classification of whether an event is FOG or not



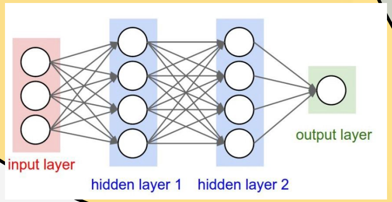
What We Learned

Technical Skills

Professional Skills



SciPy



Acknowledgements

Thank you to:

Our Coach Harshini Donepudi for her constant guidance and communication throughout the entire process, ensuring we stayed on track and met all milestones!

And Advisors: Seth Haney, Barbara Marebwa for their expert guidance, technical insight, and unwavering encouragement!


Q
A

Thank you