

Presented By:

- Lupinetti Filippo
- Paride Russo

July 22, 2025

ECG: SAFE OR NOT?



Binary ECG Classification: Normal vs. Anomalous

WHY THIS STUDY?

The electrocardiogram (**ECG**) is the main non-invasive tool to monitor the electrical activity of the heart and detect arrhythmias, ischemia, or other abnormalities.

Automating the classification between 'normal' and 'abnormal' traces allows us to:

- Reduce reporting times
- Support cardiologists by reducing workload
- Improve the timely detection of at-risk patients

Our work fits into this context, proposing a deep learning model to assist clinicians and ensure more consistent diagnostic outcomes."



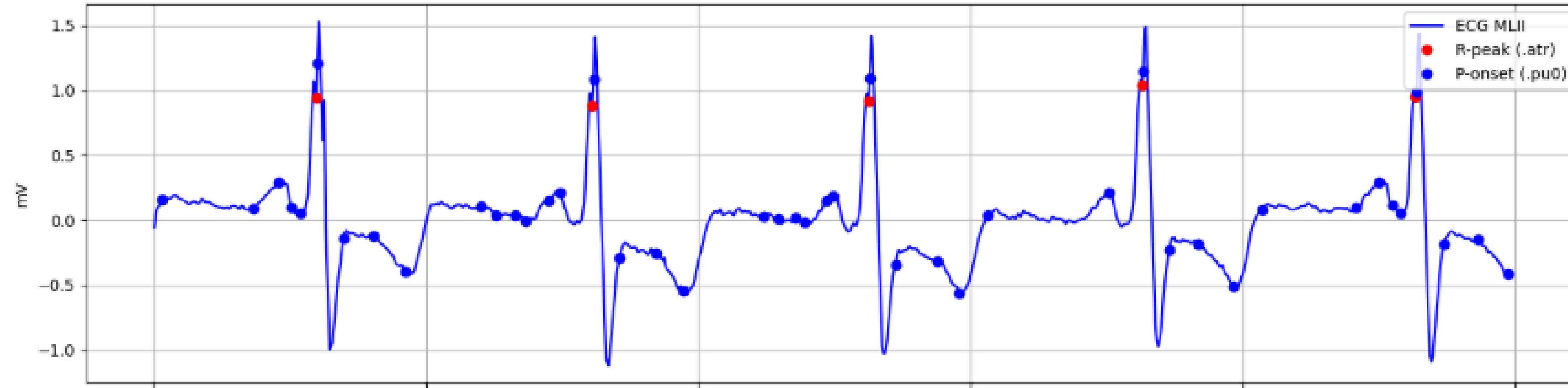
QT - DATABASE

- **105** patients included in the database
- 2 leads: **MLII** (Modified Lead II) & **V1** (Precordial lead)
- **15** minutes per signal, sampled at 250 Hz (~225,000 samples per lead)
- File Types (up to 9 per record):
 - *.hea* → Header file (metadata)
 - *.dat* → ECG signals
 - *.atr*, *.man*, *.qtl*, *.qt2*, *.qlc*, *.q2c*, *.pu*, *.pu0*, *.pul* → Annotations (manual & automatic)

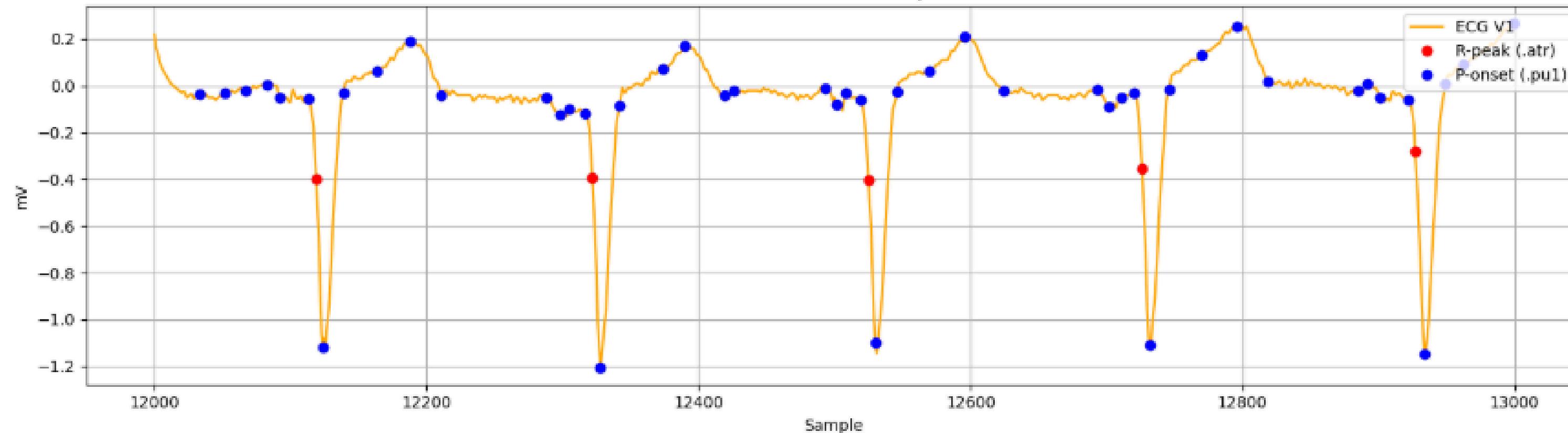
For our analysis we mainly focused on ***.dat***, ***.atr*** and ***.pu*** files.

VISUALIZATIONS

Record sel853 - MLII with .atr and .pu0 annotations



Record sel853 - V1 with .atr and .pul annotations



DATA PREPARATION

- **Record removal**

For the patients (105) the file .atr was not always available (24 missing) so we removed those ones. (**81**)

- **Beat Annotation Symbols (.atm file)**

N → Normal beat (**N**)

All other letters (e.g., L, R, A, etc.) → Abnormal Beats (**A**).

Non-letter symbols (such as /, +, ~, ", and |) do not represent beats and are therefore ignored.

Record: sel104

--- Original (.atr) ---

Unique symbols: ['+', '/', 'N', 'V', 'f']

Total annotations: 1113

First 6 symbols: ['/', '/', '/', '/', '/', 'f']

--- Filtered (.atm) ---

Unique symbols: ['A', 'N']

Total annotations: 349

First 6 filtered symbols: ['A', 'A', 'A', 'A', 'A', 'A']

Symbol	Description	Label to Use
N	Normal beat	Normal
L	Left bundle branch block beat	Abnormal
R	Right bundle branch block beat	Abnormal
A	Atrial premature beat	Abnormal
a	Aberrated atrial premature beat	Abnormal
J	Nodal (junctional) beat	Abnormal
S	Supraventricular premature beat	Abnormal
V	Premature ventricular contraction (PVC)	Abnormal
F	Fusion of ventricular and normal beat	Abnormal
e	Atrial escape beat	Abnormal
j	Nodal escape beat	Abnormal
f	Fusion of paced and normal beat	Abnormal
s	Supraventricular escape beat	Abnormal
Q	Unknown beat	Abnormal
/	Non-beat marker (e.g., segment boundary)	Ignore
+	Rhythm change annotation	Ignore
~	Signal artifact or noise	Ignore
"	Comment (not a beat)	Ignore
	Isolated marker	Ignore

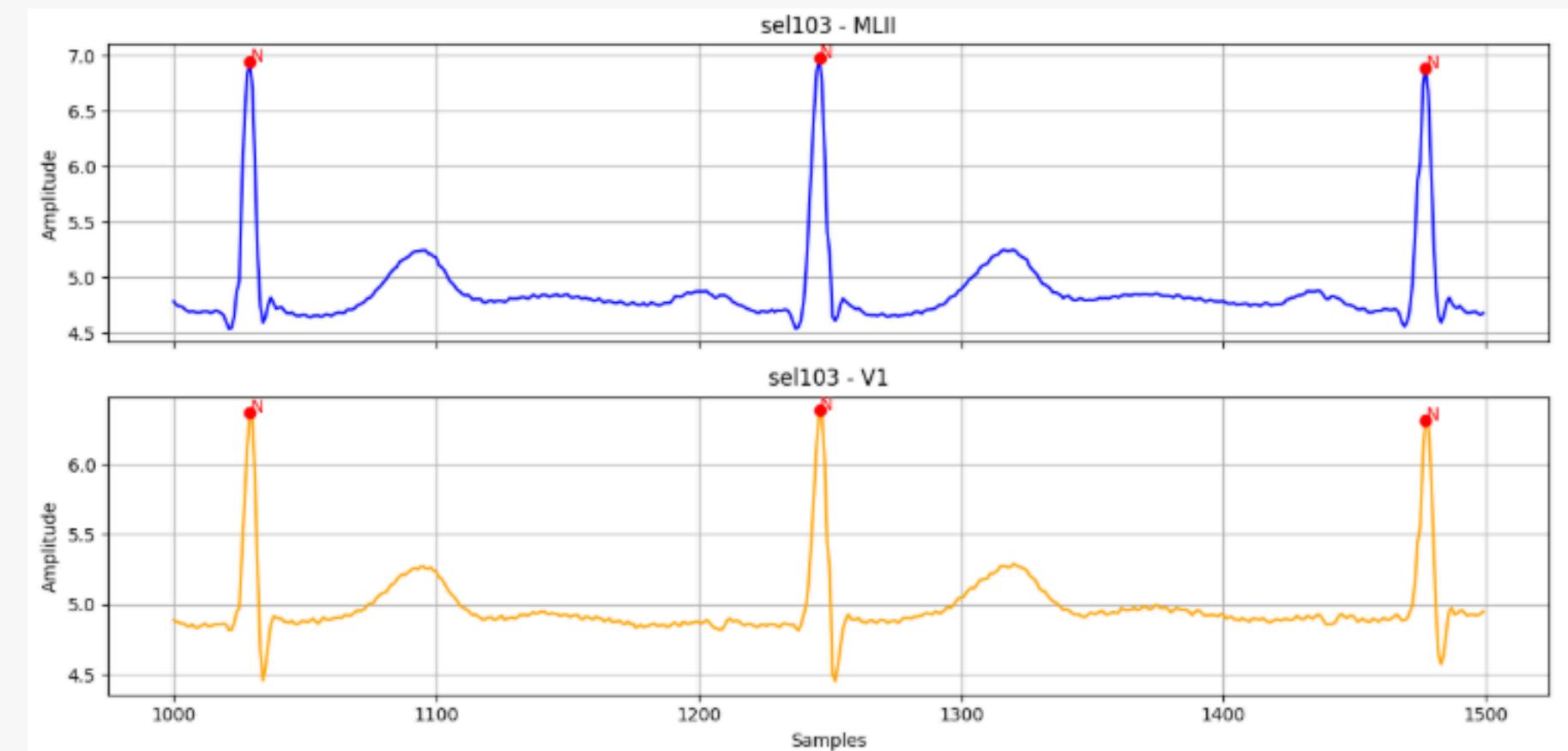
• Windows creation

In this step, we segment each ECG recording into 2-second windows and assign a label to each window based on the R-peaks it contains.

For each window:

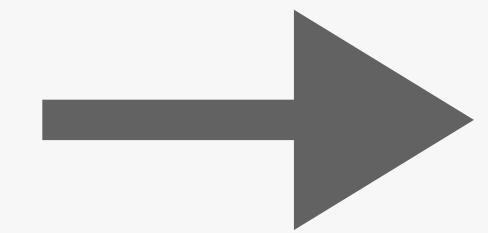
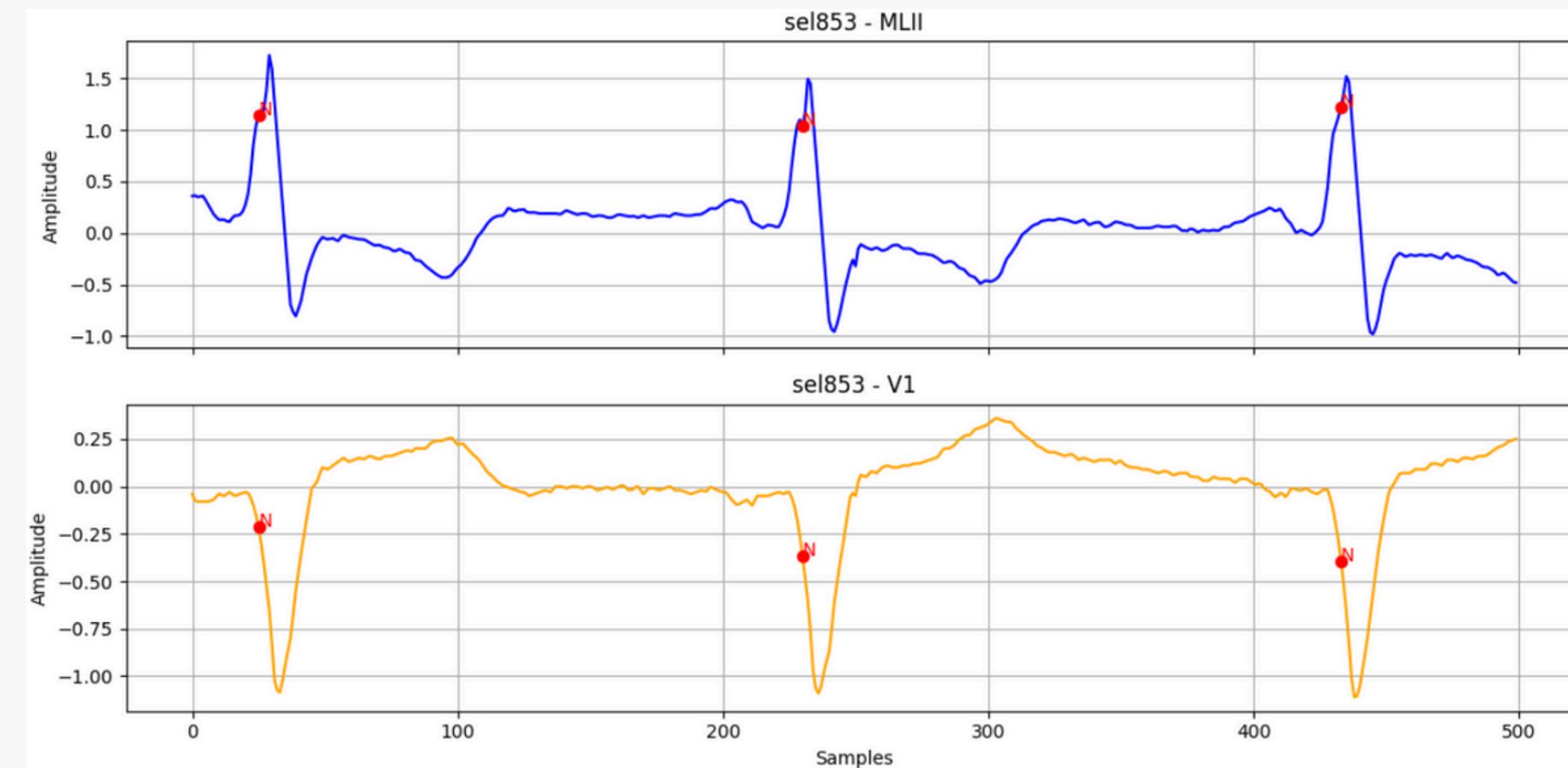
- We detect the R-peaks using .atr annotations.
- We match each R-peak to its corresponding label ('N' for normal, or any other symbol marked as abnormal) from the .atm file.
- If at least one abnormal beat is found in the window, the entire window is labeled as Abnormal; otherwise, it is labeled as Normal.

	record	window_start	window_end	label	r_peaks
0	sel102	54000	54500	Abnormal	[54125, 54265, 54475]
1	sel103	0	500	Normal	[14, 184, 399]
2	sel103	500	1000	Normal	[608, 819]
3	sel103	1000	1500	Normal	[1029, 1246, 1477]
4	sel103	1500	2000	Normal	[1697, 1905]
5	sel103	2000	2500	Normal	[2113, 2324]
6	sel103	2500	3000	Normal	[2532, 2745, 2973]
7	sel103	3000	3500	Normal	[3200, 3429]
8	sel103	3500	4000	Normal	[3649, 3875]
9	sel103	4000	4500	Normal	[4091, 4316]
10	sel103	4500	5000	Normal	[4550, 4772, 4984]



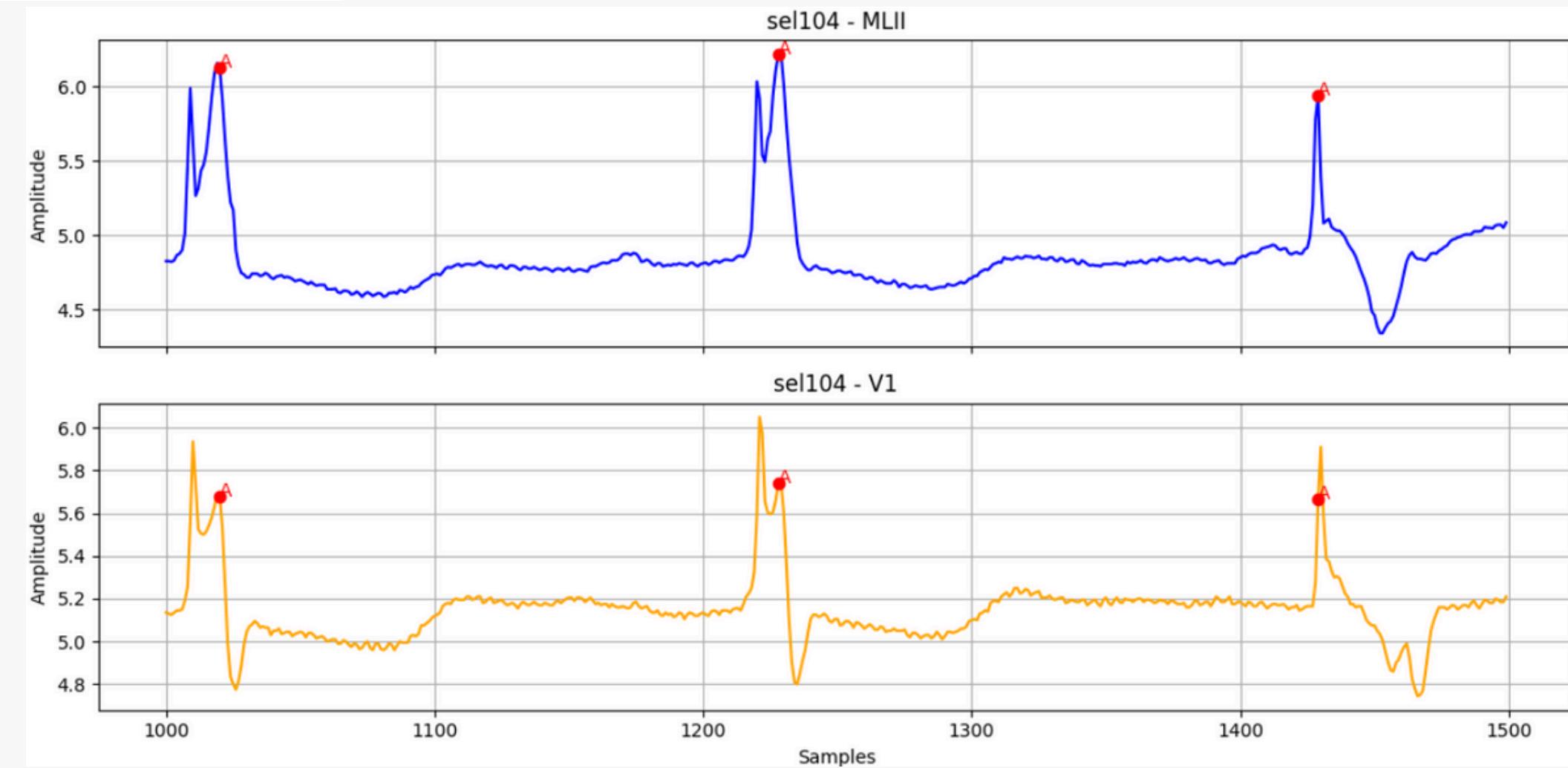
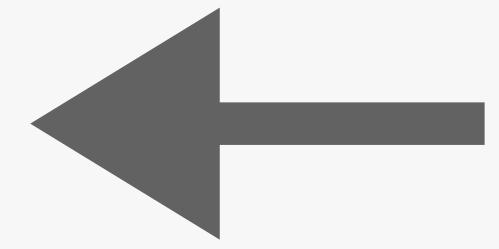
R-peak positions: [1029, 1246, 1477]
Labels of the peaks: ['N', 'N', 'N']

Final label for this window: Normal



Label: N

Label: A



DATASET SPLITTING

label		
Normal		32343
Abnormal		3347

To properly evaluate the model's performance and avoid data leakage, the dataset is divided into three subsets:

1. Training Set (80%)

This subset is used to train the model, helping it learn patterns in the data. The model's parameters are updated based on this data.

2. Validation Set (10%)

This subset helps tune model hyperparameters and monitor generalization performance.

3. Test Set (10%)

This portion will be used only for final evaluation and remains untouched during training and validation.

Z-SCORE NORMALIZATION on ECG Signals:

- on **Training Data**:
 - Normalization for each 2-second window (500 samples)
 - Performed separately for each channel (MLII and V1)
 - Normalization parameters (mean and standard deviation) calculated on the training set
- Applied to **Validation** and **Test** Data
 - Training set normalization parameters applied to validation and test sets for consistency
- **Outcome**:
 - Each window has:
 - **Mean** = 0
 - **Standard deviation** = 1



Why normalize?

- ECG amplitudes can vary significantly between patients or due to noise.
- Z-score normalization ensures all windows are on a comparable scale.
- This helps the model focus on signal shape and patterns, rather than absolute amplitude.

Normalization improves model convergence and robustness across heterogeneous signals.

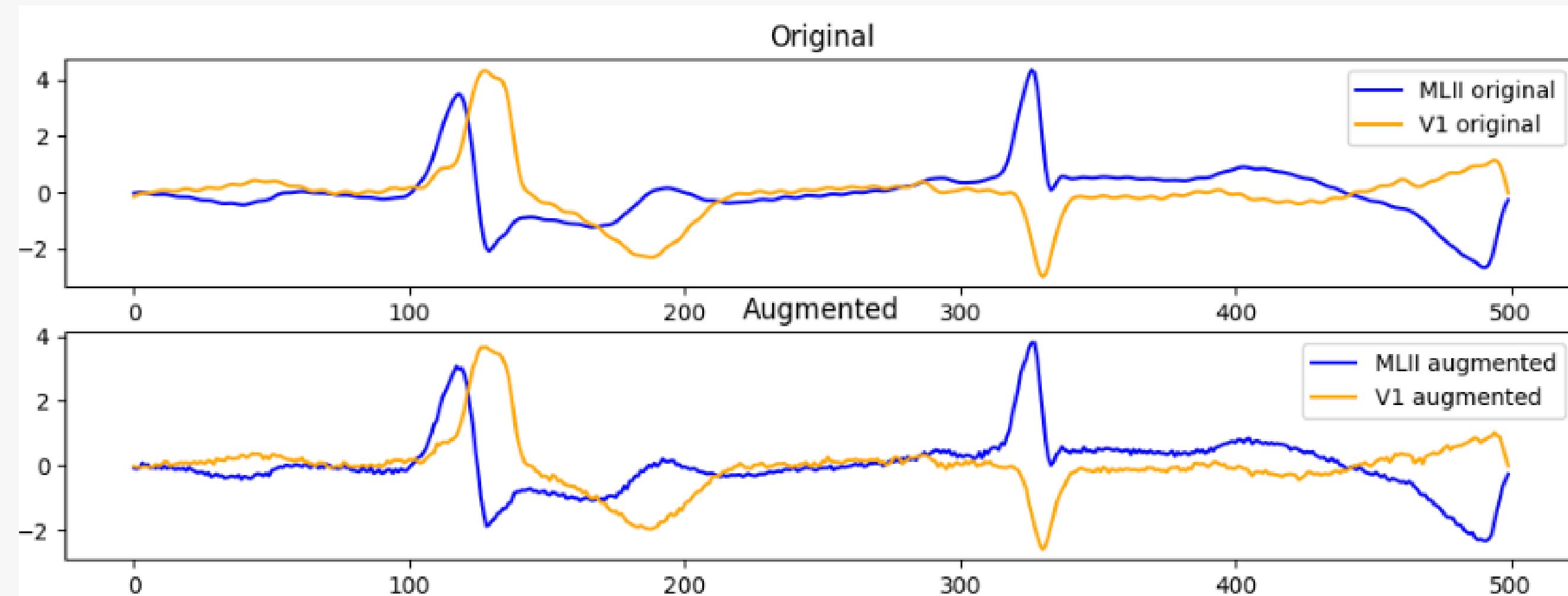
Training Set Balancing

The original dataset was highly imbalanced, with a large majority of normal beats compared to abnormal ones. To address the imbalance in the training set, we perform two key steps:

1. **OVERSAMPLING** with **Data Augmentation**

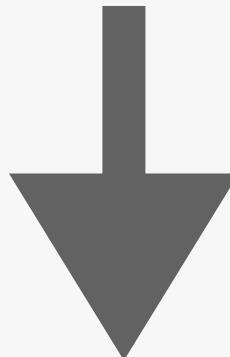
We generate synthetic abnormal samples by applying random transformations to the original abnormal windows:

- **Jittering**: Adding small Gaussian noise to simulate sensor variability.
- **Scaling**: Randomly amplifying or attenuating the signal to introduce realistic variation.



2. **UNDERSAMPLING** cutting normal windows

We performed **random undersampling** of the normal beats to match the number of abnormal samples.



The final training set is **perfectly balanced**, containing an equal number of normal and abnormal windows (50% each).

This strategy helps prevent bias during model training and ensures fair learning across classes.

```
Original Abnormal: 2692
Original Normal: 26220
Total abnormal: 2692 original + 8076 augmented = 10768
Balanced training set: 21536 total | Normal=10768, Abnormal=10768
```

```
Final balanced training distribution:
-> Normal: 10768 windows (50.00%)
-> Abnormal: 10768 windows (50.00%)
Total windows: 21536
```

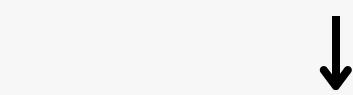
HYPERPARAMETER OPTIMIZATION

To improve model performance, we used Keras Tuner to automatically search for the best CNN configuration.

Key steps:

- **Input shape:** ECG windows of **500 samples × 2 channels**
- **Hyperparameters optimized:**
 - Number of convolutional filters
 - Kernel size
 - Dense layer units
 - L2 regularization strength
 - Dropout rate
 - Learning rate
- **Optimization objective:** **Maximize validation recall** (focus on minimizing false negatives)
- **Search method:** **Random Search** over **30 trials** with early stopping.

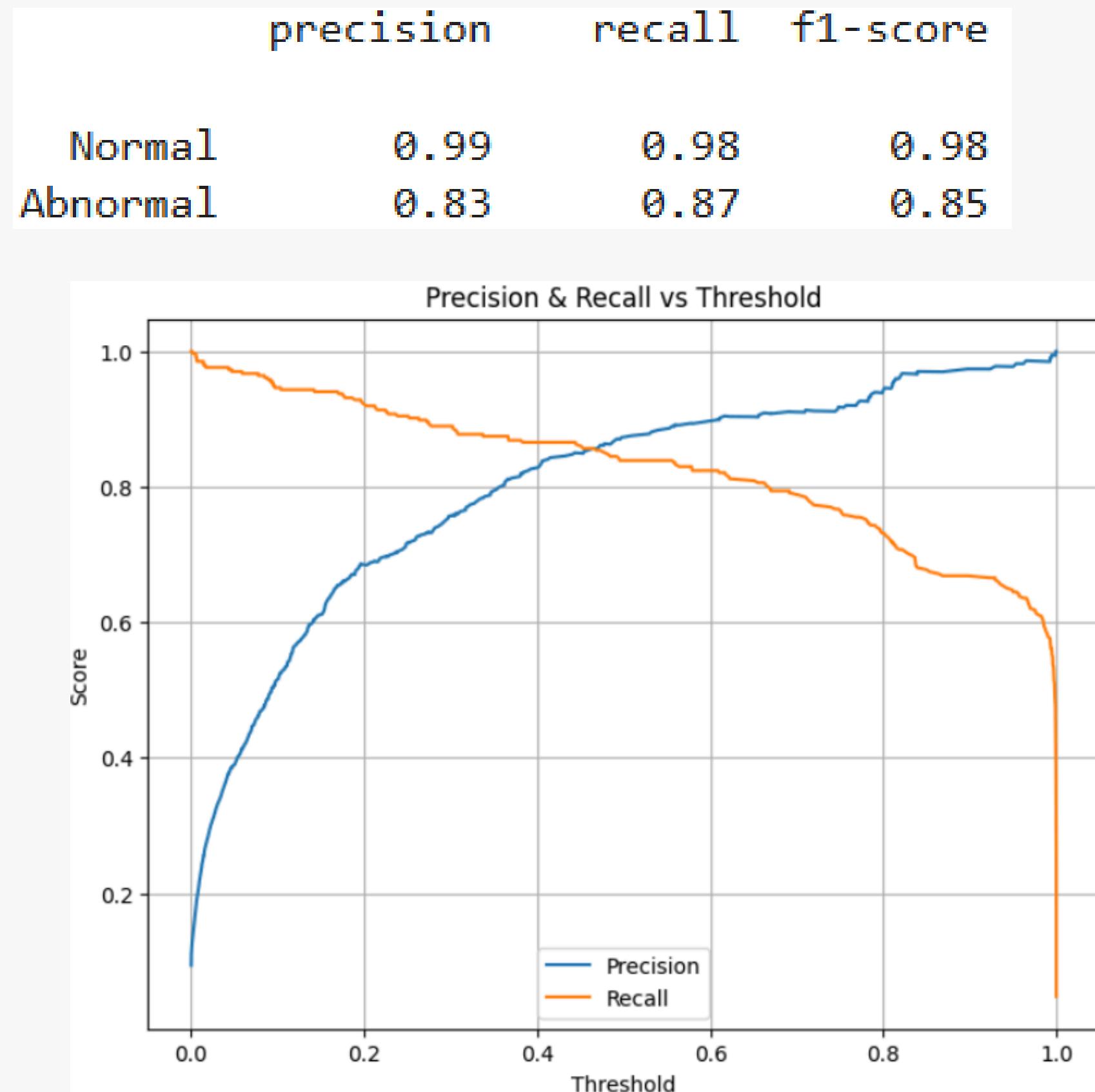
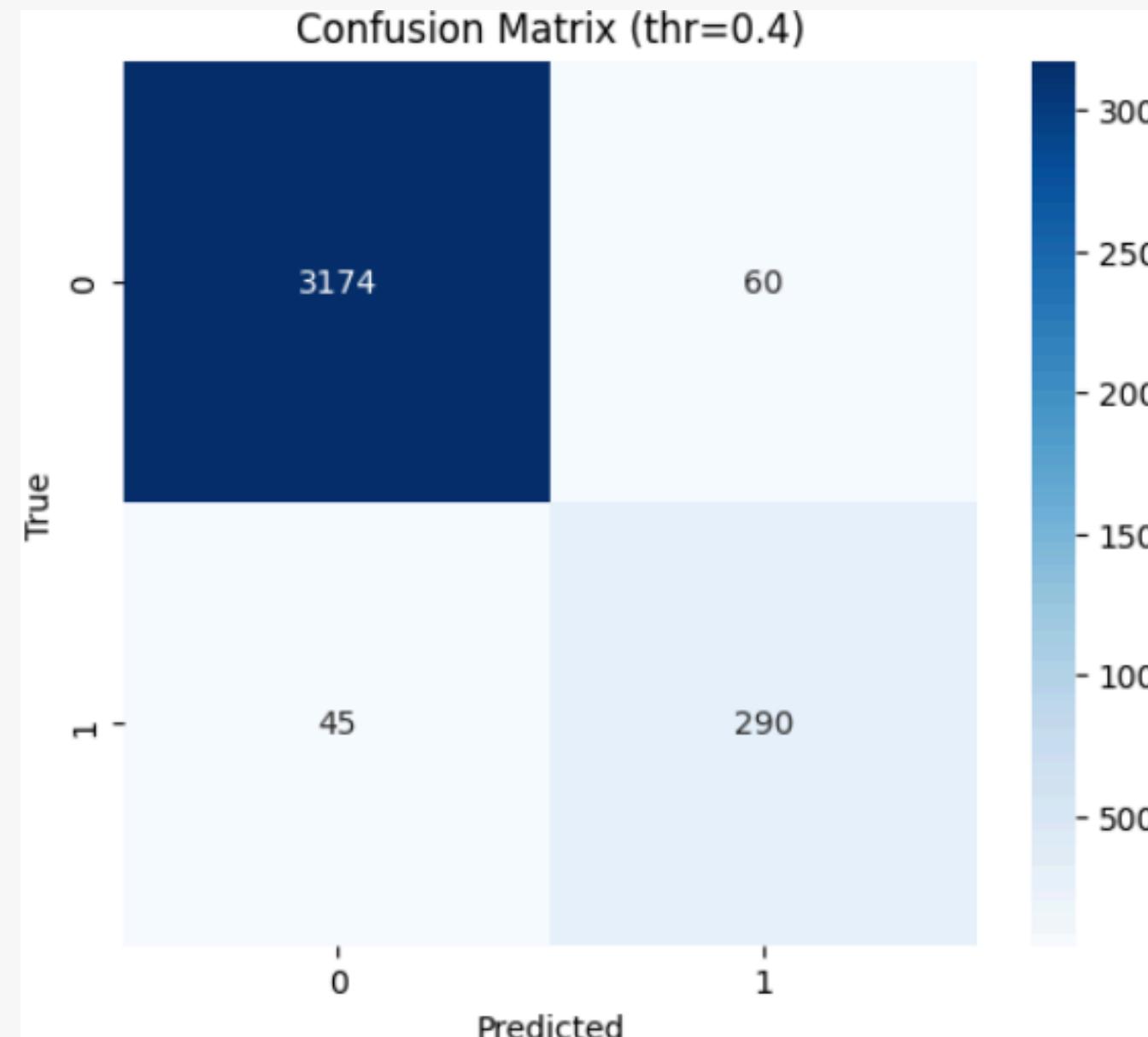
input_shape = (500, 2)



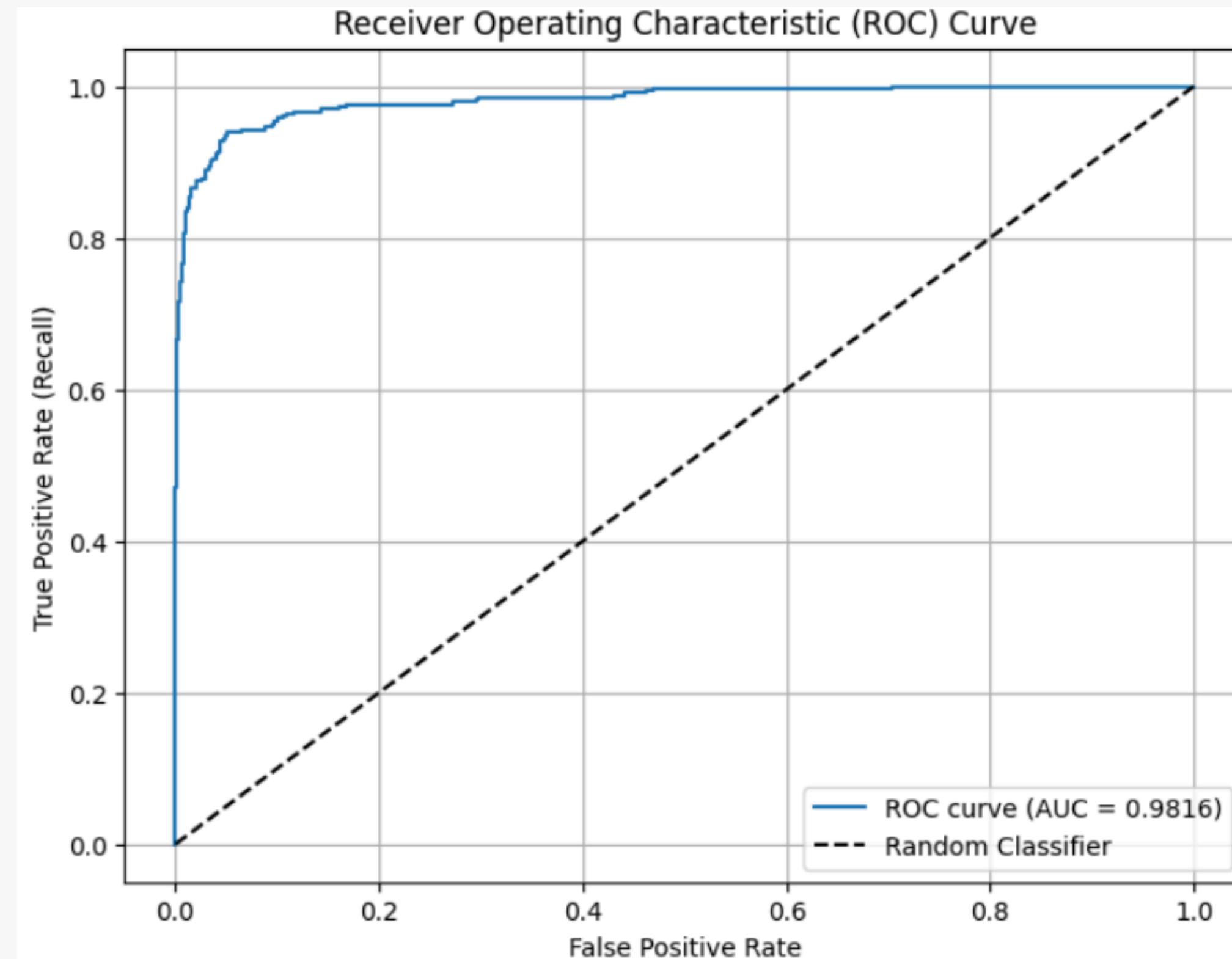
1D CNN - STRUCTURE



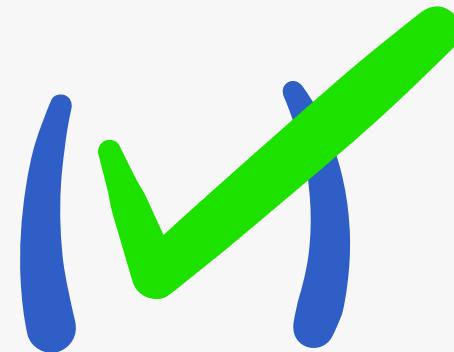
MODEL EVALUATION: TEST SET RESULTS



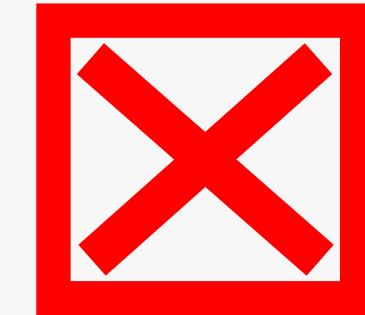
ROC - CURVE



ADVANTAGES AND DISADVANTAGES of the study



- Dual-lead ECG input with expert-verified labels
- Targeted data augmentation instead of simple duplication
- Hyperparameter tuning for optimal performance
- Evaluation using both precision and recall
- Ground truth derived from cardiologist annotations

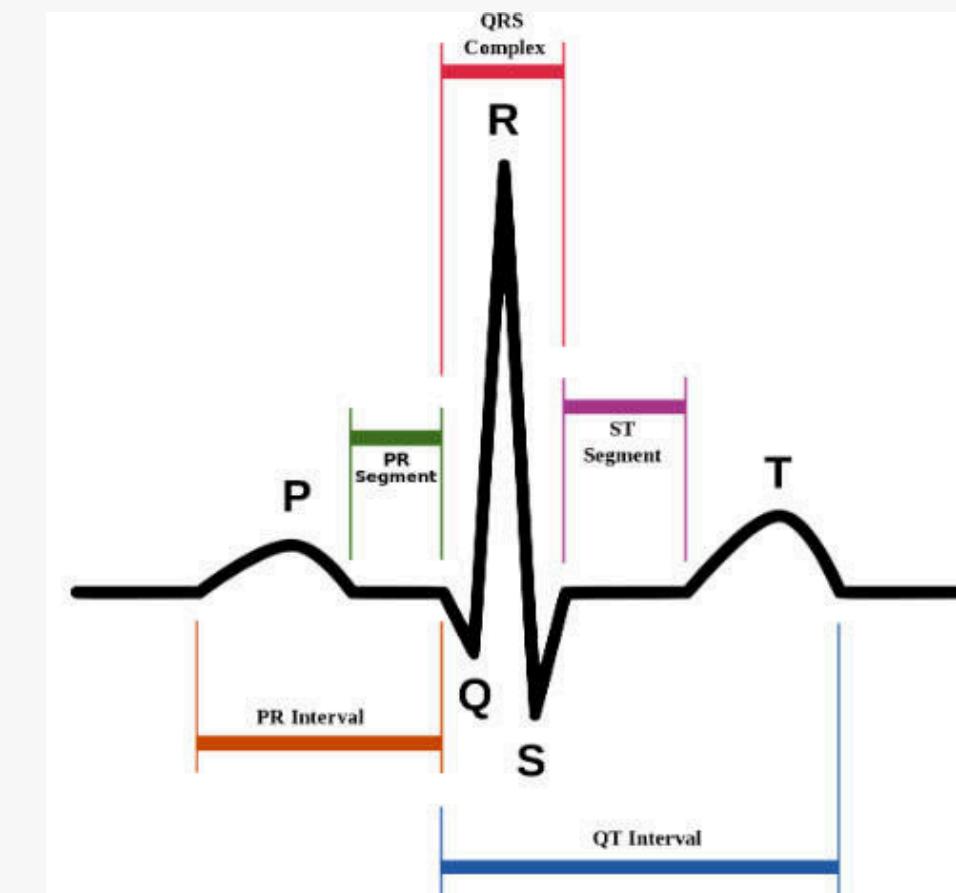


- Limited number of anomalous samples
- Fixed windows spanning only 2–3 heartbeats
- Binary classification; lacks multiclass anomaly differentiation

FUTURE DEVELOPMENTS?

- **Multiclass classification**

Differentiate specific arrhythmia types (e.g., extrasystoles, fibrillation, tachycardia) instead of just “normal vs. abnormal.”



- **Using CWT-based scalograms**

Convert ECG segments into time-frequency scalogram images and feed them to a CNN, explicitly capturing non-stationary arrhythmia features (e.g., fibrillation bursts, prolonged pauses) that pure time-domain methods miss.

