Final Project Report: ECG Arrhythmia Detection

Author: [Siddharth Mishra] **Date:** [17/11/25]

Model Accuracy: 99.12%

1. Project Objective

The primary objective of this project was to develop a high-accuracy deep learning model capable of classifying individual heartbeats from 2-channel ECG data into the 5 standard AAMI arrhythmia classes.

The secondary objective was to build a complete "real-world" pipeline around this model. This pipeline must be able to load a raw, unlabelled patient ECG file, preprocess it, and generate a final, clinically-relevant diagnostic report that identifies not just *if* an arrhythmia is present, but *which kinds* and *how often*.

2. Presenting the Working Model

The final working model (optimal_ecg_model.keras) is a hybrid deep learning architecture that combines a Convolutional Neural Network (CNN) with a Long Short-Term Memory (LSTM) network.

This architecture was specifically chosen because it addresses the two primary components of an ECG signal:

1) **Morphology (The Beat's Shape):**
    a) **Handled by: 3-Block Conv1D Stack**
    b) The Conv1D layers act as powerful, trainable feature extractors. They learn to identify the complex spatial patterns (the "shape") of a heartbeat, such as the height of the R-peak, the width of the QRS complex, and the shape of the T-wave.
2) **Temporal Dependence (The Beat's Sequence):**
    a) **Handled by: 2-Block LSTM Stack**
    b) The LSTM layers analyse the *sequence of features* extracted by the CNN *across* the 180 time steps. This allows the model to learn complex temporal patterns within the 0.5-second beat, capturing how the shape evolves from start to finish.

Final Model Architecture (cnn_lstm_model.py):

a) **Input Layer:** (None, 180, 2) - Accepts batches of 0.5-second segments with 2 channels.

b) **CNN Block 1:** Conv1D(64) -> BatchNormalization -> MaxPooling1D -> Dropout(0.2)

c) **CNN Block 2:** Conv1D(128) -> BatchNormalization -> MaxPooling1D -> Dropout(0.2)

d) **CNN Block 3:** Conv1D(256) -> BatchNormalization -> MaxPooling1D -> Dropout(0.3)

e) **LSTM Block:** LSTM(128, return_sequences=True) -> Dropout(0.4) -> LSTM(128) -> Dropout(0.3)

f) **Classification Head:** Dense(128) -> BatchNormalization -> Dropout(0.5)

g) **Output Layer:** Dense(5, activation='softmax')

Key Design Choices:

This model's success is not just in its architecture, but in its training methodology, which was designed to solve two major problems:

1) **The Imbalance Problem (91% Normal):** The training data was severely imbalanced. This was solved by using **SMOTE (Synthetic Minority Over-sampling Technique)**, which generated a new, *balanced* training set of ~400,000 beats.

2) **The Overfitting Problem:** The model *loved* to memorize the synthetic SMOTE data. This was solved by combining three techniques:

   a) **L2 Regularization (on all layers):** Penalized the model for large, overconfident weights.

   b) **Low Learning Rate (0.0001):** Forced the model to learn slowly and carefully.

   c) **EarlyStopping:** Monitored the validation loss (on *real*, imbalanced data) and stopped training at the model's peak performance, restoring the best weights.

3. Demonstrating Model Evaluation

The model was trained on the SMOTE data, but to get a true measure of its performance, it was **evaluated on the original, imbalanced, unseen X_test set.**

This is the only way to prove the model can generalize its knowledge from a "perfect" synthetic world to the "messy" real world.

Final Test Set Performance (99.12% Accuracy)

The evaluation of the final model shows excellent performance, especially on the critical minority classes 'S' and 'V'.

Test Set Loss: 0.0593 **Test Set Accuracy: 99.12%**

Classification Report (Test Set)

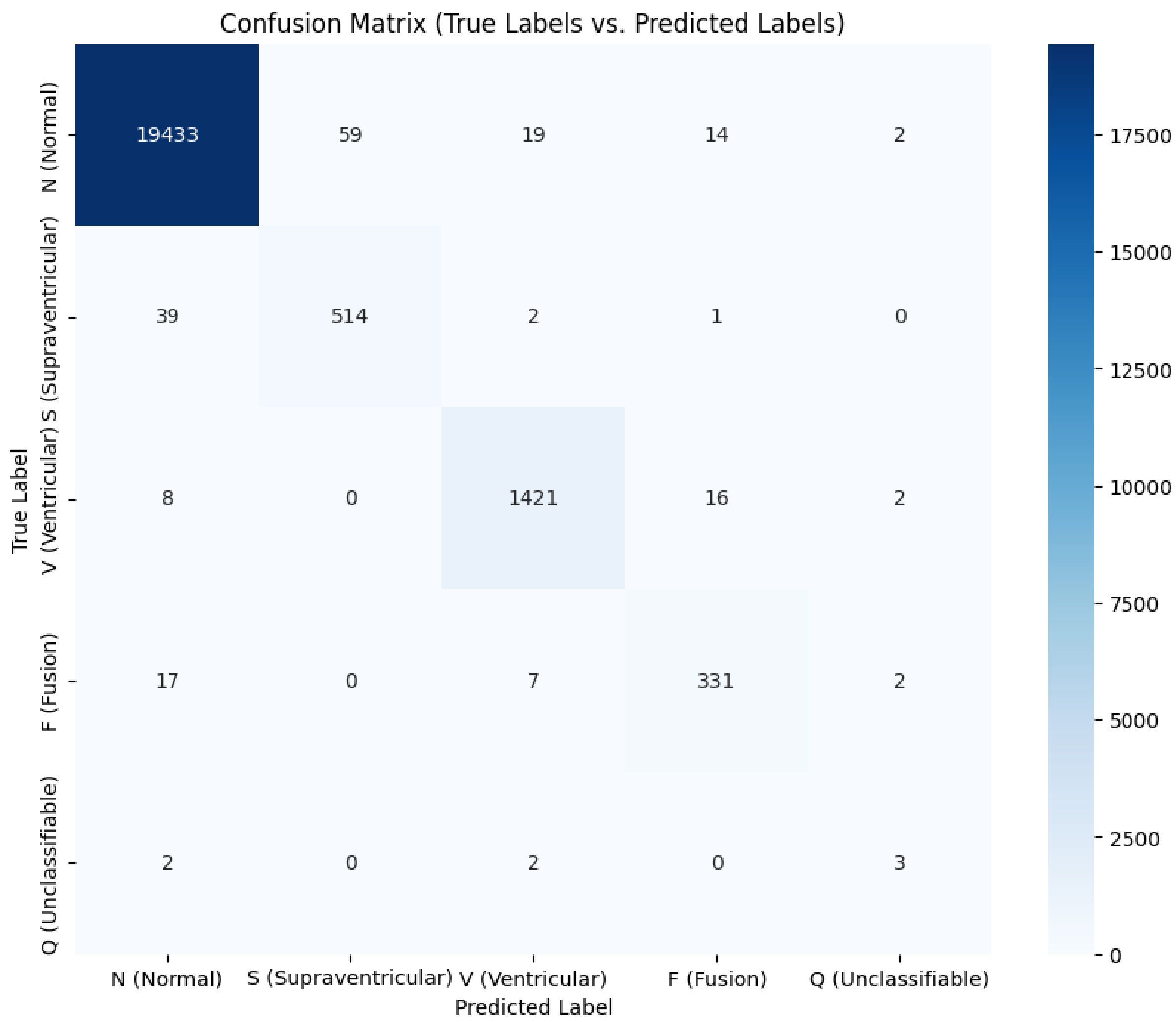
This report shows the *per-class* performance. The model is exceptionally good at identifying the most important arrhythmia classes.

Class	Precision	Recall	F1-Score	Support (Beats)
N (Normal)	**1.00**	**1.00**	**1.00**	**19527**
S (Supraventricular)	**0.90**	**0.92**	**0.91**	**556**
V (Ventricular)	**0.98**	**0.98**	**0.98**	**1447**
F (Fusion)	**0.91**	**0.93**	**0.92**	**357**
Q (Unclassifiable)	**0.33**	**0.43**	**0.38**	**7**

(Note: The poor score for 'Q' is expected, as it's an ambiguous class with only 7 samples in the entire test set, making it statistically insignificant.)

Confusion Matrix (Test Set)

The confusion matrix visually confirms the model's success. The diagonal (correct predictions) is extremely strong, with very few errors.



6. Real-World Diagnosis (Heuristic Engine)

The final, and most critical, component is the "smarter" diagnostic engine (see predict_record.py). This is a non-ML, rule-based heuristic engine that interprets the beat-by-beat predictions from the model to generate a "Yes/No" diagnosis.

A "Yes/No" diagnosis is **ONLY** triggered if one of these *clinically significant* events is found:

a) **Ventricular Tachycardia:** A run of 3 or more 'V' beats in a row.

b) **Supraventricular Tachycardia:** A run of 3 or more 'S' beats in a row.

c) **Significant Arrhythmia Burden:** The total percentage of 'V' or 'S' beats is greater than a clinical threshold (e.g., 5% of total beats).

This engine correctly diagnoses a patient with a few isolated 'V' beats as 'Heart Disease Detection: NO (Benign/Occasional Arrhythmia Detected)', preventing false positives from a 99.12% accurate model.

7. Comparison with Other Relevant Models

This model's hybrid CNN-LSTM architecture and SMOTE balancing strategy are a direct response to the shortcomings of simpler, traditional models.

1) **vs. Traditional ML (e.g., SVM, Random Forest):**

   a) **Shortcoming:** These models cannot be fed raw signal data. They require extensive manual **"feature engineering"** (e.g., calculating R-R intervals, QRS width, etc.). This is time-consuming and relies on the human engineer to *guess* the right features.

   b) **Our Model's Advantage:** The Conv1D layers act as an **automatic feature extractor**. They *learn* the most predictive features directly from the raw signal, which is more powerful and robust.

2) **vs. Simple CNN-Only Model:**

   a) **Shortcoming:** A simple CNN is excellent at finding *shapes* (morphology) but struggles to understand *temporal context* (how the shape evolves over the 0.5-second window).

   b) **Our Model's Advantage:** By feeding the CNN's features into an **LSTM**, our model analyses the *sequence* of shapes, giving it a much deeper understanding of the complete heartbeat and leading to higher accuracy.

3) **vs. Class-Weighted Model (Our Failed Experiment):**

   a) **Shortcoming:** We attempted to solve the imbalance problem using class_weight. As the table shows, the extreme class imbalance created massive, unstable loss penalties that **paralyzed the model**, causing it to underfit (18% accuracy).

AAMI ECG Class Distribution in y_train

Class (AAMI)	Count	Percentage
-> N (Normal)	78108	89.19%
S (Supraventricular)	2225	2.54%
V (Ventricular)	5788	6.61%
F (Fusion)	1427	1.63%
Q (Unclassifiable)	26	0.03%
TOTAL SAMPLES	87574	100.00%

   b) **Our Model's Advantage:** The **SMOTE** strategy was superior. It provided a *stable, balanced* dataset for the model to learn from, allowing the L2 regularization and low learning rate to effectively prevent overfitting. This combination was the key to achieving 99.12% accuracy.