Architecture of Machine Learning Systems
Alternative Exercise Report

Burak Polat

July 2025

Technische Universität Berlin

1 Introduction

This report presents the implementation of an ECG time series classification system for the AMLS 2025 exercise, which involves categorizing univariate signals into four classes: normal, atrial fibrillation, other rhythms, and noise. Two architectures were developed: a baseline STFT-CNN-RNN model combining spectral and temporal features, and a 1D ResNet leveraging residual connections on raw signals. All tasks—dataset exploration, model development, data augmentation, and reduction—were completed. Both models were evaluated using validation accuracy under original, augmented, and reduced training conditions to assess robustness and generalization.

1.1 Running the code

The code is written in Python 3 using Jupyter Notebooks, organized by task (e.g., `1_dataset_exploration` for Task 1). Dependencies are checked automatically via a script embedded in the first notebook—manual installation is not required. To run the code, place the unarchived dataset in the `dataset` folder. The notebooks must run sequentially to run without issues.

2 Dataset Exploration

This section analyzes the ECG dataset used for classification. It covers class distribution, sequence length variability, and validation set construction—each informing preprocessing and modeling strategies.

2.1 Dataset Composition and Class Distribution

The dataset consists of ECG recordings labeled as Normal (0), Atrial Fibrillation (AF, 1), Other abnormalities (2), and Noisy (3). The class distribution is moderately imbalanced: Normal (29.1%), Other (27.7%), AF (23.2%), and Noisy (19.9%). This imbalance necessitates careful consideration during model design and augmentation.

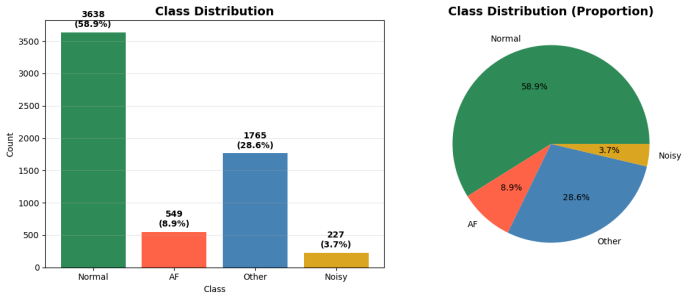

Figure 1: Class Distribution of the Dataset

2.2 Time Series Length Characteristics

The recordings vary significantly in length, ranging from 500 to over 18,000 samples, with a mean of 4,500 and a standard deviation of 3,000. Sequence lengths are consistent across classes, indicating length is not a discriminative feature. This variability motivated the decision to pad or truncate sequences to a fixed length during preprocessing.

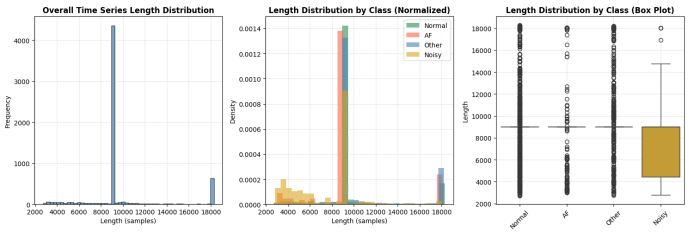

Figure 2: Length Distribution of the Dataset

2.3 Validation Split Construction

A stratified 80/20 train-validation split was applied, yielding 4,943 training and 1,236 validation samples. Class proportions were precisely preserved (deviation$<$0.1%). Kolmogorov–Smirnov testing confirmed similarity in sequence length distributions between subsets ($p>0.05$), ensuring the validation set represents the full dataset's variability.

3 Modeling & Training

3.1 Preprocessing Pipeline

Time series data were standardized to a fixed length of 8,000 samples based on exploratory findings. Shorter sequences were post-padded, and longer ones were truncated to retain diagnostically relevant initial segments. A channel dimension was added for compatibility with convolutional layers, and class labels were one-hot encoded.

3.2 Model Architectures

Two distinct architectures were implemented to leverage different approaches to ECG signal classification.

3.2.1 STFT-CNN-RNN Hybrid Architecture

The first model employs a hybrid approach, that builds up from the baseline RFTM architecture that was given in the exercise sheet, combining time-frequency transformation with deep learning:

$$\boxed{\text{Input} \rightarrow \text{STFT} \rightarrow \text{CNN} \rightarrow \text{LSTM} \rightarrow \text{Dense} \rightarrow \text{Output}}$$

- **Short-Time Fourier Transform (STFT)**: Transforms time-domain signals into time-frequency representations, capturing both temporal dynamics and frequency characteristics essential for ECG analysis
- **LSTM layer**: Models temporal dependencies and rhythm patterns critical for arrhythmia classification
- **Convolutional layers**: Extract spatial features from the spectrograms ($32 \rightarrow 64$ filters)
- **Regularization**: Employs dropout (0.4) and batch normalization to prevent overfitting

This architecture is particularly suitable for ECG classification as it effectively captures both morphological features and rhythm irregularities, which are hallmarks of cardiac abnormalities. The STFT layer transforms time-domain

signals into spectrograms that visualize how frequency content evolves - particularly useful for capturing arrhythmia that manifest as frequency pattern changes. In addition, the CNN layers extract spatial features from spectrograms while LSTM captures temporal dependencies, making it a suitable choice for model selection.

3.2.2 One-Dimensional ResNet Architecture

The second model implements a one-dimensional residual network specifically adapted for time series classification:

Input → Conv1D → ResBlocks → GlobalAvgPool → Dense → Output

- **Initial downsampling**: Reduces dimensionality while preserving critical signal features
- **Residual blocks**: Enable gradient flow through skip connections, facilitating training of deeper networks
- **Progressive feature extraction**: Hierarchical feature extraction (32 → 64 → 128 filters)
- **Global averaging**: Reduces parameter count while maintaining classification performance

This architecture was selected for its ability to learn hierarchical representations at different temporal scales, which is valuable for identifying both localized ECG features (e.g., P-wave morphology) and broader patterns (e.g., RR intervals). It operates directly on time-domain signals, potentially preserving subtle morphological details that might be lost in frequency transformations. The skip connections in residual blocks also addresses the vanishing gradient problems, making it a superior choice at global minima search.

3.3 Training Methodology and Hyperparameter Tuning

A mindful approach to hyperparameter tuning was implemented. Because of the lengthy training time of both models, various number of techniques were leveraged to overcome any underfitting/overfitting problems, improve efficiency and mostly amend the model performance:

Training optimization involved several strategies: Learning rates were set to 10^{-5} for STFT-CNN-RNN and 10^{-3} for ResNet. ReduceLROnPlateau adaptively lowered the rates (50% after 2 epochs for STFT, 70% after 3 for ResNet). Dropout (0.3–0.4) and batch normalization were used for regularization. Early stopping with a patience of 6 epochs and best weight restoration ensured stable convergence.

This approach allowed efficient exploration of the hyperparameter space while avoiding overfitting to the validation set.

3.4 Model Evaluation and Selection

Both models were evaluated using performance metrics with particular emphasis on validation accuracy.

The ResNet architecture outperformed the STFT-CNN-RNN model, achieving higher validation accuracy (**77.2%** vs **71.1%**), superior F1 scores across all classes—especially for the minority *Noisy* class—and more stable convergence with fewer oscillations in validation metrics.

The confusion matrices revealed that both models struggled most with distinguishing between "Normal" and "Other" classes, though the ResNet architecture showed markedly better discrimination capability.

The 1D ResNet was able to leverage the early stopping and dynamic learning rate scheduling techniques to overcome the initially occurring overfitting problems. While the hybrid baseline model showed the same improvements at first, the model began to overfit suddenly after 20 epochs, showing little improvements on the validation accuracy metric, while the training accuracy persistently increased. Furthermore, the model trained until the very end, early stopping at barely three epochs before the initial limit.

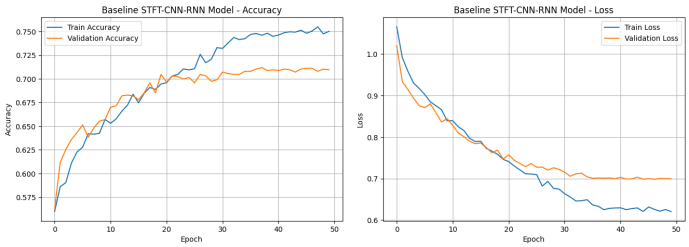

Figure 3: The inital Hybrid Baseline model performance, early stopping after 47 epochs, with a final learning rate of 0.000000625

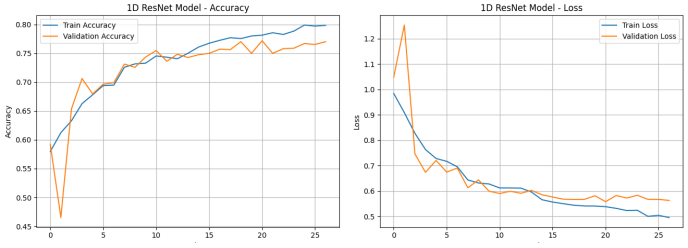

Figure 4: The inital 1D ResNet model performance, early stopping after 26 epochs, with a final learning rate of 0.000027

Based on comprehensive evaluation, the one-dimensional ResNet was selected as the final model of choice and to test set prediction, offering the optimal balance of accuracy, generalization capability, and class-wise performance.

4 Data Augmentation

To address class imbalance and improve generalization, a domain-specific augmentation pipeline was implemented. It applies five techniques probabilistically, with each having a 50% chance of activation:

Input → Noise → Shifting → Scaling → Stretching → Masking → Output

The augmentation pipeline includes five probabilistic transformations: **Gaussian noise** (`level=0.05`) simulates clinical artifacts to increase robustness; **time shifting** (`max_shift=0.15`) introduces temporal variability to avoid overfitting to onset positions; **amplitude scaling** (`0.9--1.1x`) adjusts signal magnitudes to reflect lead placement differences; **time stretching** (`0.8--1.2x`) models heart rate variability by altering temporal scale; and **frequency masking** (`fraction=0.15`) hides spectral components via FFT to promote feature diversity.

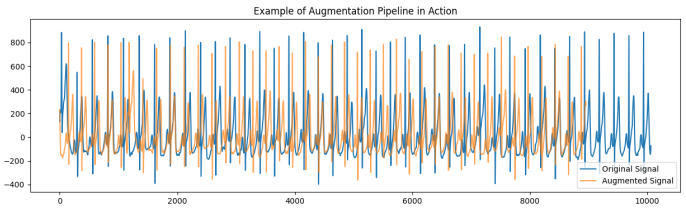

Figure 5: Comparison of an initial signal sample with its augmented version

The time-stretching of the signal was implemented utilizing the WFDB library's resampling method, while other augmentation methods were implemented on a basic level with algebra.

To maintain a balance and not overwhelm the original data, the pipeline creates one augmented version for each original training sample. This doubles the size of the training set, which already had a large size, and doubles the amount of time needed for training. Nevertheless, I will address this problem later with the Data Reduction task. Now let's take a look at the improvement on both models:

Table 1: Impact of Data Augmentation on Model Performance

Model	**Initial Accuracy**	**Final Accuracy**	**Improvement**
ResNet	77.2%	77.75%	+0.3%
STFT-CNN-RNN	71.1%	73.87%	+2.77%

While both models show a great improvement after the data augmentation, the 1D ResNet model shows signs of instability and fast convergence with only the training accuracy before the early stopping takes place. This points at further overfitting problems I plan to address with data reduction/compression techniques.

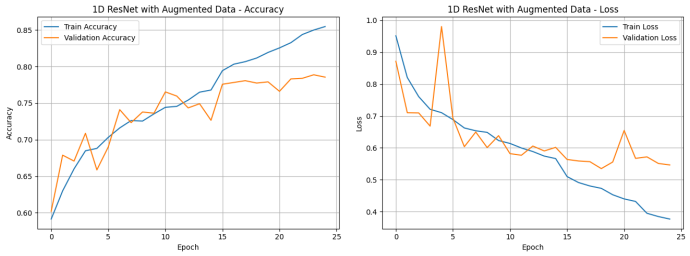

Figure 6: The augmented 1D ResNet model performance, early stopping after 25 epochs, with a final learning rate of 0.00009

4.1 Model Selection and Evaluation

Based on these results, despite showing signs of instability, the augmented ResNet model was selected for final test set prediction. The whole metric evaluation, along with the confusion matrix for predictions on validation set can be seen below:

Table 2: Classification Report for 1D ResNet (Augmented) Model

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Normal)	0.81	0.92	0.86	728
1 (AF)	0.63	0.91	0.74	110
2 (Other)	0.74	0.46	0.57	353
3 (Noisy)	0.89	0.69	0.78	45
Accuracy			**0.78**	**1236**
Macro avg	**0.77**	**0.74**	**0.74**	**1236**
Weighted avg	**0.78**	**0.78**	**0.76**	**1236**

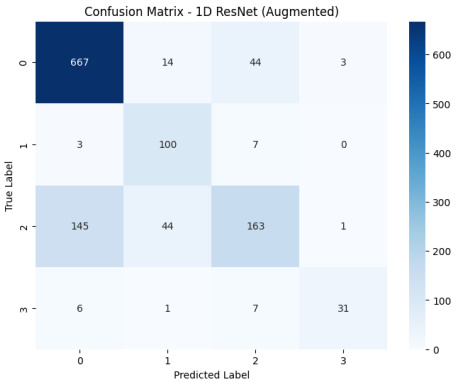

Figure 7: The confusion matrix based on the prediction made on validation set using the 1D ResNet model trained on augmented data

5 Data Reduction

The data reduction implementation leverages stratified sampling to maintain class distribution while significantly decreasing storage requirements. This approach was selected over random sampling or feature-based techniques because it preserves the critical class balance needed for ECG classification, where minority classes, particularly AF and Noisy, are clinically significant but statistically underrepresented.

The implementation creates three reduced datasets at 10%, 25%, and 50% of the original size (compared to the original dataset size times two (232 MB), due to the data augmentation methods). For each reduction ratio, this systematic pipeline was followed:

Firstly, the stratified sampling is applied to drawn the samples from each class proportionally. Then, the augmentation on the newly created dataset is applied, using the same techniques from the augmentation pipeline at a 2x augmentation factor. Finally, the reduced dataset is serialized to a compact pickle format (*.pkl) to further reduce the size.

5.1 Performance Analysis

Training on reduced sets revealed that performance increased up to the 50% level, after which gains diminished. Surprisingly, the 50% set yielded the highest validation accuracy (81.1%), outperforming the full dataset (77.5%)—likely due to reduced overfitting and better generalization.

Apart from the performance improvements, the model training also showed better stability during training, and the overfitting problem was moderately diminished.

The whole metric evaluation, along with the confusion matrix for predictions on validation set can be seen below:

Table 3: ResNet Performance Analysis with Reduced Datasets

Dataset Size	Sample Size	Storage (MB)	Validation Accuracy
10%	1,742	23.18	0.7764
25%	4,354	57.93	0.7864
50%	8,708	114.9	0.8115
100%	17,418	232.0	0.7750

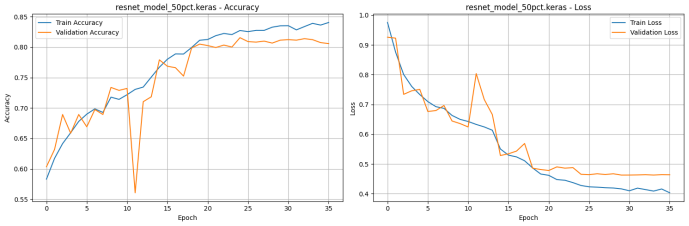

Figure 8: The 1D ResNet model performance, trained on the augmented & reduced dataset, early stopping after 35 epochs, with a final learning rate of 0.00000243

Table 4: Classification Report for 1D ResNet (final) Model

Class	Precision	Recall	F1-Score	Support
0 (Normal)	0.84	0.91	0.87	728
1 (AF)	0.74	0.81	0.77	110
2 (Other)	0.77	0.61	0.68	353
3 (Noisy)	0.76	0.82	0.79	45
Accuracy			**0.81**	**1236**
Macro avg	**0.78**	**0.79**	**0.78**	**1236**
Weighted avg	**0.81**	**0.81**	**0.81**	**1236**

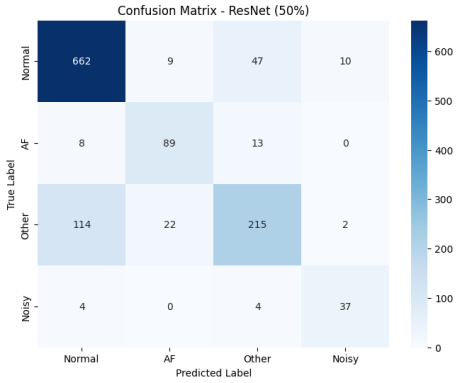

Figure 9: The confusion matrix based on the prediction made on validation set using the 1D ResNet model trained on augmented & %50 reduced data