Comparing CNN, Recurrent, Hybrid, and Transformer Models for Multi-Label ECG Classification on PTB-XL

Huey Phan
University of Texas at Austin
Austin, Texas, USA
hp22929@eid.utexas.edu

Abstract

Electrocardiograms (ECGs) are widely used to detect cardiovascular abnormalities, but manual ECG interpretation requires clinical expertise and is time-consuming. This project investigates whether deep learning models can accurately classify diagnostic ECG superclasses from 12-lead waveforms using the PTB-XL dataset. Five PyTorch models were compared: 1D CNN, LSTM, GRU, CNN-LSTM, and Transformer Encoder. The task was treated as multi-label classification using five PTB-XL diagnostic superclasses: normal, myocardial infarction, ST/T change, conduction disturbance, and hypertrophy. Models were evaluated using macro precision, macro recall, macro F1-score, micro F1-score, and macro ROC-AUC. The best model was a 1D CNN trained for 30 epochs with a learning rate of 5e-4, achieving a macro F1-score of 0.718 and macro ROC-AUC of 0.916. These results suggest that convolutional models are effective for learning local ECG waveform patterns and may provide a lightweight approach for automated ECG classification.

CCS Concepts

• **Computing methodologies** → **Neural networks**; • **Applied computing** → **Health informatics**.

Keywords

ECG classification, deep learning, multi-label classification, PTB-XL, convolutional neural networks

1 Introduction

Cardiovascular disease remains a major public health problem, and electrocardiograms are one of the most common diagnostic tools used to evaluate cardiac function. ECGs provide information about heart rhythm, conduction, ischemic changes, and structural abnormalities. However, ECG interpretation requires clinical expertise, and automated ECG classification may help support clinical screening and decision-making.

Deep learning has become increasingly useful for biomedical signal analysis because it can learn features directly from raw signals. In ECG analysis, convolutional neural networks can identify local waveform morphology, recurrent models can capture sequential dependencies, and Transformer-based models can learn long-range signal relationships. However, model performance can vary depending on architecture, learning rate, and class imbalance.

This project applies deep learning to multi-label ECG diagnostic superclass classification using the PTB-XL dataset. The goal is to compare five model architectures: 1D CNN, LSTM, GRU, CNN-LSTM, and Transformer Encoder. The main research question is: Which deep learning architecture performs best for classifying PTB-XL ECG diagnostic superclasses from raw 12-lead ECG signals?

The main contributions of this project are: (1) implementing five PyTorch deep learning models for raw ECG classification, (2) comparing performance across multiple learning rates and epoch settings, and (3) identifying the best-performing lightweight model for multi-label ECG superclass classification.

2 Related Work

The PTB-XL dataset [9] is a large public 12-lead ECG dataset designed for machine learning research. It contains clinical ECG records and diagnostic labels that can be grouped into diagnostic superclasses such as NORM, MI, STTC, CD, and HYP. The dataset also provides recommended stratified folds for model evaluation, which helps make experiments more reproducible.

Previous work has used PTB-XL as a benchmark for automated ECG classification. [7] evaluated deep learning approaches on PTB-XL and found that convolutional neural network architectures, especially ResNet- and Inception-style models, performed strongly for ECG analysis tasks. This supports the use of CNN-based models as a baseline and comparison point in this project.

Recent PTB-XL studies have also explored simplified CNN-based approaches [5], data-centric preprocessing, and class balancing. Some work has noted that minority classes, especially hypertrophy, can be harder to classify because of class imbalance and overlapping ECG patterns. This motivates the use of macro F1-score and per-class evaluation in addition to overall performance metrics.

Beyond PTB-XL specifically, deep learning has also been applied to ECG interpretation at clinical scale. [2] trained a deep neural network on single-lead ambulatory ECG recordings and showed that it could match or exceed cardiologist-level performance for arrhythmia detection, demonstrating that convolutional architectures can learn clinically meaningful waveform features directly from raw signal. [6] trained a residual CNN on a large 12-lead ECG dataset to automatically diagnose six types of abnormalities and validated it against certified cardiologists, showing that deep learning can generalize to multi-abnormality 12-lead classification at scale. These results motivate treating raw-waveform deep learning as a practical approach for the multi-label superclass classification task addressed in this project, rather than relying on hand-engineered ECG features.

3 Data

This project used PTB-XL dataset, a public dataset of 12-lead ECG recordings [9]. Each ECG record is approximately 10 seconds long and includes diagnostic SCP codes. The metadata file ptbxl_database.csv was used to locate ECG waveform files and retrieve diagnostic labels. The file scp_statements.csv was used to map diagnostic SCP codes into diagnostic superclasses.

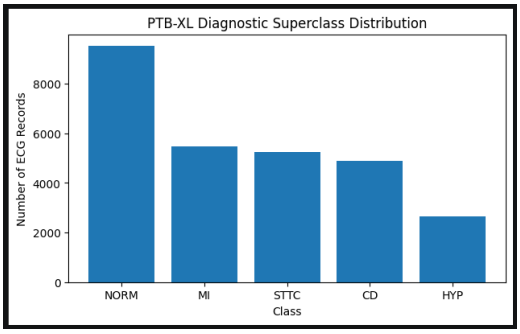

Figure 1: Distribution of PTB-XL diagnostic superclasses after filtering to records with at least one of the five target labels.

The classification task used five diagnostic superclasses: NORM, MI, STTC, CD, and HYP. NORM represents normal ECGs, MI represents myocardial infarction, STTC represents ST/T changes, CD represents conduction disturbance, and HYP represents hypertrophy. Since one ECG can have more than one diagnostic label, this project treated the task as multi-label classification.

The PTB-XL strat_fold column was used for data splitting. Folds 1–8 were used for training, fold 9 was used for validation, and fold 10 was used for testing. This follows the dataset's recommended fold-based evaluation setup.

Figure 1 shows the number of ECG records associated with each diagnostic superclass after filtering out records with no superclass label. NORM is the majority class, while HYP has substantially fewer records than the other four classes. Because a single ECG can carry more than one diagnostic label, the counts shown do not sum to the total number of records. This class imbalance motivated the use of positive class weighting during training (Section 4.4) and macro-averaged metrics during evaluation (Section 4.2).

4 Methodology

4.1 Models

The five model architectures were selected to compare different approaches for ECG signal learning.

1D CNN [5] uses three convolutional blocks. The first applies 32 filters with kernel size 7 and padding 3, the second applies 64 filters with kernel size 5 and padding 2, and the third applies 128 filters with kernel size 3 and padding 1. Each convolutional layer is followed by a ReLU activation, and the first two blocks are followed by max pooling with a kernel size of 2. The final feature map is reduced with adaptive average pooling, passed through a dropout layer (rate 0.3), and mapped to the 5 output classes with a linear layer.

LSTM (Long Short-Term Memory) [7] is a single-layer bidirectional LSTM with an input size of 12 and a hidden size of 64 per direction. The final forward and backward hidden states are concatenated into a 128-dimensional vector and passed through a linear layer that maps to the 5 output classes.

GRU (Gated Recurrent Unit) [1] uses the same configuration as the LSTM model: a single-layer bidirectional GRU with an input size of 12 and a hidden size of 64 per direction, followed by a linear layer that maps the concatenated 128-dimensional hidden state to the 5 output classes.

CNN-LSTM combines convolutional feature extraction with recurrent temporal modeling. Two convolutional blocks (64 and 128 filters, kernel sizes 7 and 5) with ReLU and max pooling first extract local waveform features, which are then passed to a single-layer bidirectional LSTM with hidden size 64 per direction. The concatenated 128-dimensional hidden state is mapped to the 5 output classes with a linear layer.

Transformer [8] first embeds the raw signal with a 1D convolution (kernel size 7, stride 4, padding 3) that projects the 12 input leads into a 64-dimensional sequence. This embedding is passed through a Transformer encoder with 2 layers, 4 attention heads, feedforward dimension 128, and dropout rate 0.2. The encoder output is mean-pooled across the sequence and mapped to the 5 output classes with a linear layer. The Transformer Encoder uses self-attention to model long-range relationships across the ECG sequence.

All models were implemented in PyTorch. Data processing was performed using pandas and NumPy [3], evaluation metrics were computed with scikit-learn, and figures were produced with Matplotlib [4].

4.2 Evaluation

Models were evaluated using macro precision, macro recall, macro F1-score, micro F1-score, and macro ROC-AUC. Macro-averaged metrics compute precision, recall, and F1 independently for each of the five diagnostic superclasses and then average the per-class scores unweighted, so that rare classes such as HYP contribute equally to the metric alongside common classes such as NORM. Micro-averaged F1, by contrast, pools true positives, false positives, and false negatives across all classes before computing a single score, which means it is dominated by the more frequent classes. Reporting both macro and micro F1 makes it possible to see whether a model is succeeding broadly across all diagnostic categories or primarily on the majority class. ROC-AUC was computed per class from the predicted probabilities and then macro-averaged, and does not depend on the choice of classification threshold.

4.3 Train/Test Splits

We used strat_fold to perform the splitting: folds 1-8 as the training set, fold 9 as the validation set, and fold 10 as the test set.

4.4 Loss

Because the task was multi-label classification, the models were trained using binary cross-entropy with logits loss. Class imbalance was handled using positive class weights computed from the training labels, so that under-represented classes such as HYP were penalized more heavily when misclassified. During evaluation, sigmoid activation was applied to model logits, and predictions were generated using a threshold of 0.5 applied uniformly across all five classes. All models were trained with the AdamW optimizer and a weight decay of 1e-4, and the model checkpoint with the highest validation macro F1-score was retained for test-set evaluation.

Figures 2–6 show the training loss curves for each model over 30 epochs.

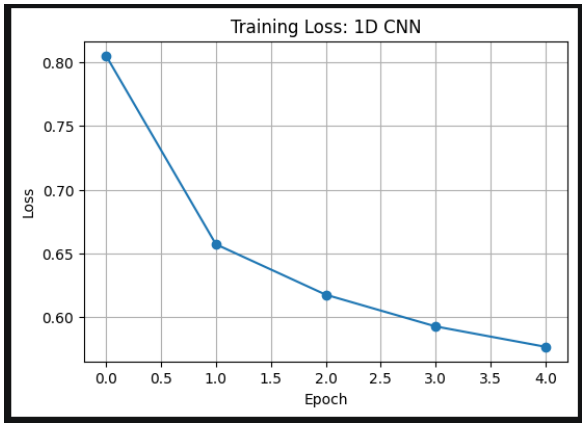

Figure 2: Training loss for the 1D CNN.

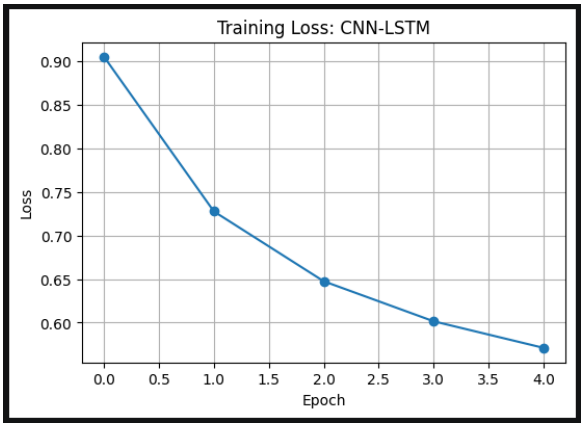

Figure 5: Training loss for the CNN-LSTM.

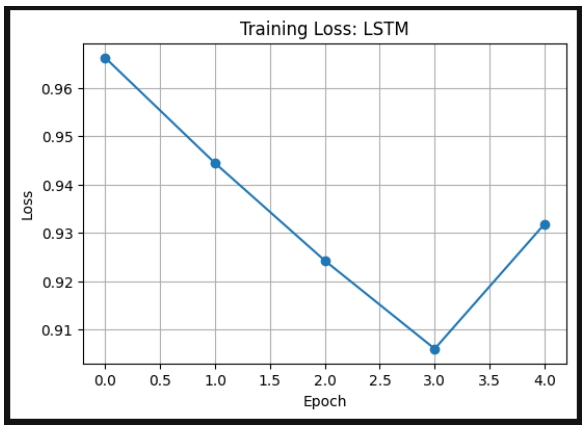

Figure 3: Training loss for the LSTM.

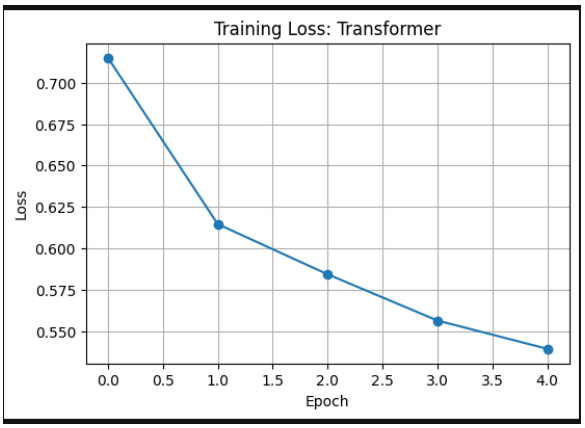

Figure 6: Training loss for the Transformer Encoder.

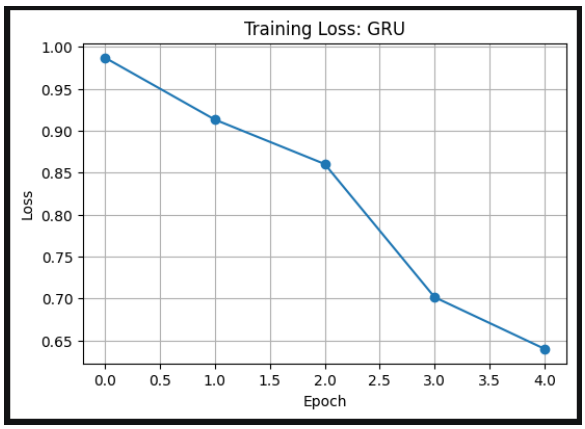

Figure 4: Training loss for the GRU.

5 Experiments and Results

Macro F1-score was selected as the primary metric for model comparison because the dataset is multi-label and class imbalance exists across diagnostic superclasses.

5.1 Best Model

The best overall model was the 1D CNN trained for 30 epochs with a learning rate of 5e-4. It achieved a macro F1-score of 0.718, micro F1-score of 0.750, and macro ROC-AUC of 0.916. CNN-LSTM was the second strongest model with a macro F1-score of 0.710, while Transformer achieved a macro F1-score of 0.694 and macro ROC-AUC of 0.899. LSTM performed worse than the CNN-based models.

5.2 Hyperparameter Tuning

Learning-rate tuning showed that model performance changed across training settings. The 1D CNN achieved its best macro F1-score using 30 epochs and a learning rate of 5e-4. Lowering the learning rate to 1e-4 generally reduced performance for CNN, LSTM, GRU, and CNN-LSTM models, suggesting that this learning rate was too small for convergence within the tested epoch budget.

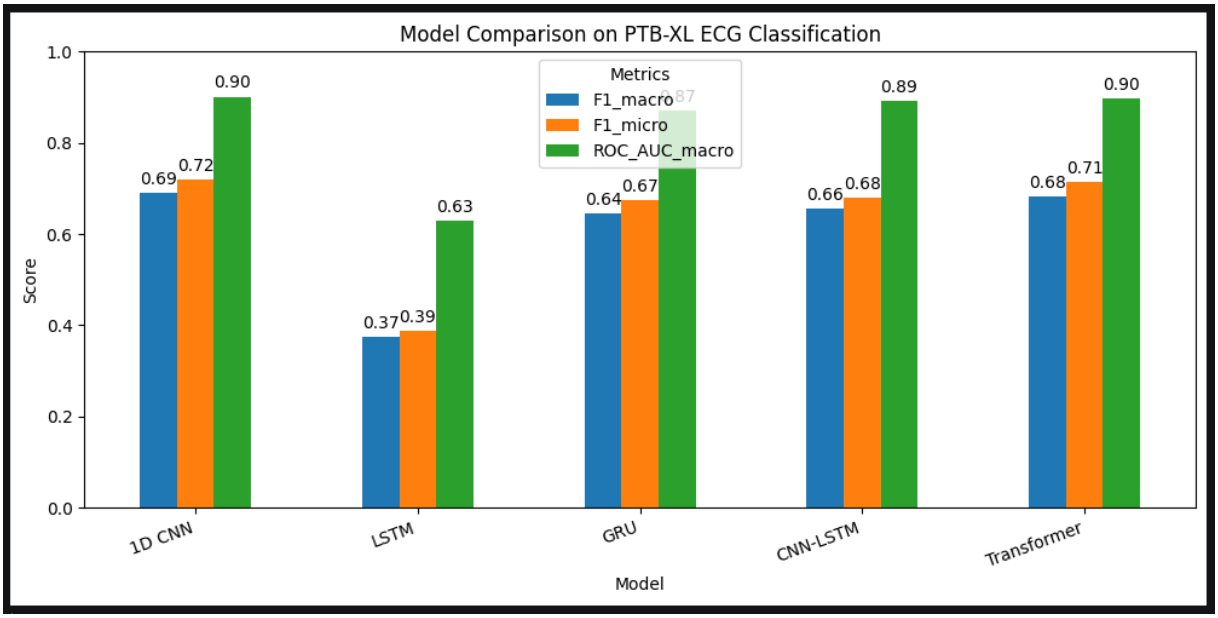

Figure 7: Comparison of macro F1, micro F1, and macro ROC-AUC across all five models on the test set.

Model	Precision Macro	Recall Macro	F1 Macro	F1 Micro	ROC AUC Macro
1D CNN	**0.647**	**0.810**	**0.718**	**0.750**	**0.916**
LSTM	0.472	0.532	0.469	0.527	0.722
GRU	0.643	0.796	0.704	0.735	0.902
CNN-LSTM	0.659	0.780	0.710	0.744	0.906
Transformer	0.638	0.766	0.693	0.727	0.899

Table 1: Test-set performance of all five architectures, trained for 30 epochs with a learning rate of 5e-4.

Setting	Best Model	F1 Macro	ROC AUC Macro
20 epochs, lr = 1e-3	1D CNN	0.708	0.918
20 epochs, lr = 5e-4	1D CNN	0.718	0.910
20 epochs, lr = 1e-4	Transformer	0.692	0.902
30 epochs, lr = 5e-4	1D CNN	0.718	0.916
30 epochs, lr = 1e-4	Transformer	0.619	0.901

Table 2: Best model performance across tested epoch and learning-rate settings.

6 Discussion

6.1 Why the CNN Model Dominates

The results show that the 1D CNN performed best overall, suggesting that local ECG waveform morphology is highly informative for diagnostic superclass classification. This makes sense because ECG abnormalities are often visible through local signal patterns such as QRS morphology, ST-segment changes, and T-wave changes. CNN-LSTM also performed well, but it did not outperform the simpler CNN model. This suggests that adding recurrent layers did not provide enough additional benefit to justify the extra complexity in this experiment.

The LSTM model performed the weakest among the tested architectures. One possible explanation is that recurrent models alone may struggle to learn local ECG morphology compared with convolutional filters. The Transformer model performed competitively, but it also did not outperform the CNN. This may be because the simple Transformer architecture used in this project was not heavily optimized, or because the dataset size and model configuration favored convolutional feature learning.

Table 3 shows the per-class classification report for the best model (1D CNN) on the test set, confirming that some classes were substantially easier to detect than others. NORM achieved the strongest performance (F1 = 0.85) and had the largest support, while HYP was the most difficult class to classify (F1 = 0.57) despite having a comparatively high recall, indicating that the model over-predicts HYP in exchange for catching more true positives. MI, STTC, and CD fell in between, with recall consistently higher than precision across every class. This pattern is consistent with the effect of positive class weighting in the loss function (Section 4.4): weighting increases recall on minority and hard-to-separate classes at the cost of precision, since the model is penalized more

Class	Precision	Recall	F1	Support
NORM	0.78	0.92	0.85	963
MI	0.69	0.78	0.73	550
STTC	0.64	0.87	0.74	521
CD	0.63	0.80	0.70	496
HYP	0.49	0.69	0.57	262
Micro avg	0.68	0.84	0.75	2792
Macro avg	0.65	0.81	0.72	2792
Weighted avg	0.68	0.84	0.75	2792

Table 3: Per-class classification report for the best model (1D CNN, 30 epochs, lr = 5e-4) on the test set, at a decision threshold of 0.5.

heavily for missing a positive label than for a false alarm. The gap between NORM and HYP performance is also consistent with the class imbalance shown in Figure 1, since HYP had the fewest labeled records among the five superclasses. The gap may also reflect genuine overlap between hypertrophy patterns and other structural or ST-T abnormalities in the underlying ECG morphology.

6.2 Limitations and Future Work

This project has several limitations. First, the models were trained only on the 100 Hz PTB-XL ECG records, not the 500 Hz records. Second, the models used a fixed threshold of 0.5 for all classes, which may not be optimal for multi-label classification. Third, the study compared relatively simple versions of each architecture and did not include stronger ECG-specific architectures such as ResNet1D or InceptionTime. Fourth, the project did not include explainability methods such as saliency maps or occlusion analysis.

Future work could improve this project by tuning class-specific thresholds, adding per-class ROC-AUC and PR-AUC, training models with multiple random seeds, and testing lead-ablation experiments. Explainability methods could also be used to highlight which parts of the ECG signal influenced model predictions.

7 Conclusion

This project compared five deep learning architectures for multi-label ECG diagnostic superclass classification using PTB-XL. The 1D CNN achieved the best overall performance, with a macro F1-score of 0.718 and macro ROC-AUC of 0.916 after training for 30 epochs with a learning rate of 5e-4. These findings suggest that convolutional models are effective for learning ECG waveform features and can outperform recurrent and simple Transformer-based models in this setting. The project demonstrates how deep learning can be applied to raw biomedical signals for healthcare classification tasks.

References

[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555 [cs.NE] https://arxiv.org/abs/1412.3555

[2] Awni Y. Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H. Tison, Codie Bourn, Mintu P. Turakhia, and Andrew Y. Ng. 2019. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine* 25, 1 (2019), 65–69.

[3] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, et al. 2020. Array programming with NumPy. *Nature* 585, 7825 (2020), 357–362.

[4] John D Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95.

[5] Keiron O'Shea and Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. arXiv:1511.08458 [cs.NE] https://arxiv.org/abs/1511.08458

[6] Antônio H. Ribeiro, Manoel Horta Ribeiro, Gabriela M. M. Paixão, Derick M. Oliveira, Paulo R. Gomes, Jessica A. Canazart, Milton P. S. Ferreira, Carl R. Andersson, Peter W. Macfarlane, Wagner Meira Jr., Thomas B. Schön, and Antonio Luiz P. Ribeiro. 2020. Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nature Communications* 11, 1 (2020), 1760.

[7] Nils Strodthoff, Patrick Wagner, Tobias Schaeffter, and Wojciech Samek. 2020. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL. arXiv:2004.13701 [cs.LG] https://arxiv.org/abs/2004.13701

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL] https://arxiv.org/abs/1706.03762

[9] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Wojciech Samek, and Tobias Schaeffter. 2022. PTB-XL, a large publicly available electrocardiography dataset. *PhysioNet* (Nov. 2022). doi:10.13026/kfzx-aw45 Version 1.0.3.