A Cascading Multi-Stage Deep Learning Approach for Detecting Chagas Disease from Electrocardiograms

Jonas Sandelin[1], Zoher Orabe[1], Ismail Elnaggar[1], Katri Karhinoja[1], Yangyang Zhao[1], Chito Patiño[1], Matti Kaisti[1], Antti Airola[1]

[1] University of Turku Finland

Abstract

Aims: For the PhysioNet Challenge 2025, our team "bug busters" developed an approach to detect Chagas disease from electrocardiograms. This parasitic infection can be life-threatening when untreated, and electocardiogram based screening could direct limited resources more efficiently.

Methods: We implemented a novel multi-stage cascading approach using five deep learning models: two ResNet18 variants with attention mechanisms, two SimpleCNN models, and an AttentionCNN. Our key innovation is a progressive filtering pipeline that ranks healthy samples by their prediction scores and removes those most confidently classified as healthy, creating increasingly focused training sets.

Results: Our approach scored 0.369 in the official stage on the validation dataset and 0.224 in the test set. Our team was ranked 14th out of 41.

Conclusion: The cascading multi-stage methodology shows promise for Chagas disease detection, overcoming the limitations of single-model approaches. Future work should investigate performance across diverse patient populations and explore interpretability of model decisions.

1. Introduction

We participated in the 2025 George B. Moody PhysioNet Challenge, which invited teams to develop automated algorithms for detecting Chagas disease from electrocardiograms (ECGs) [1, 2]. Our entry, team "*bug busters*", focused on addressing the extreme class imbalance in the training data by developing a cascading multi-stage framework that progressively filtered majority-class samples before classification. This approach enabled us to train multiple complementary neural network models and combine them in a meta-ensemble. In the following sections, we describe the details of our methodology, present the results on the official Challenge datasets, and discuss the implications and limitations of our approach.

2. Methods

2.1. Data and Preprocessing

The challenge dataset combined three sources: the CODE-15% dataset [3] with over 340,000 ECGs and self-reported Chagas labels, the SaMi-Trop dataset [4] with 1,631 serologically confirmed positive cases, and the PTB-XL dataset [5] with 21,799 presumed negative recordings. [6]

ECG signals were resampled at 400 Hz and set to a fixed length of 2048 samples by randomly segmenting the full-length recordings, with the shorter signals zero-padded as needed. This length was chosen as 2048 samples provide sufficient information about cardiovascular health. While longer segments may improve arrhythmia detection, it was not the focus of this work. Random segment selection during training exposed the models to different parts of each recording, thereby improving generalization. For testing, a multiclip approach was used, in which predictions from five randomly selected segments of the same recording were averaged to improve robustness.

The signals were standardized per lead by subtracting the mean and dividing by the standard deviation of each individual lead. The signals were not filtered since in our internal testing we noticed it decreasing performance.

Internal validation was performed on a dataset of 150,000 ECG recordings, comprising all available SaMi-Trop and PTB-XL data, supplemented with Chagas positive cases and non-duplicate healthy samples from CODE-15. This optimization reduced computational requirements and sped up the testing process. The complete training and evaluation of the model required approximately 6 hours and 1 hour, respectively, while with the full dataset it took days. We estimated that including all available Chagas data would not drastically affect testing and developing.

2.2. Multi-Stage Progressive Filtering

We implemented a cascading training approach to address the significant class imbalance in the dataset. This

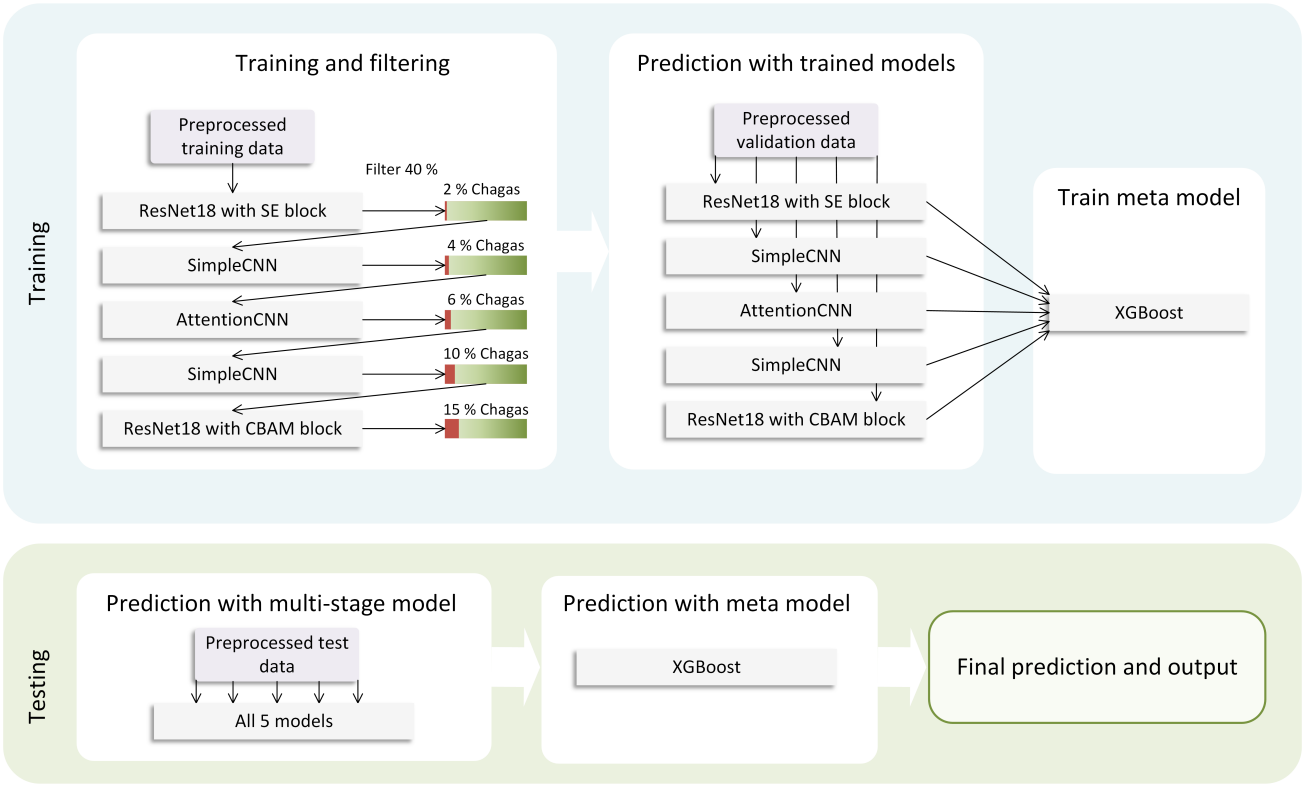

Figure 1. Overview of the proposed cascading multi-stage pipeline for training and testing. (a) Preprocessing and data split: Raw 12-lead ECG signals are standardized and divided into 80% training and 20% validation sets. (b) Progressive filtering model: A sequence of five neural networks (Stage 1–5) are trained. After every stage, 40% of the healthy samples classified with the highest confidence are removed. (c) Meta-ensemble and prediction: The outputs from all five stages are given to an XGBoost meta-model, which integrates the stage-level predictions into a final classification score.

methodology, inspired by the work of Sadreddin et al. [7], progressively filters majority class samples (negative cases) to create increasingly balanced training sets. By reducing the dominance of the majority class, we enable the models to better detect subtle patterns indicative of Chagas disease. Previous studies have established that mitigating class imbalance leads to improved classification performance [7, 8]. Our approach cascades through the dataset 5 times, roughly filtering out 85% of the healthy data.

2.2.1. Pipeline

Our proposed framework comprises three primary components: a data preprocessing stage, a multi-stage progressive filtering model, and an ensemble meta-model. This pipeline is shown in Figure 1. The dataset was divided into 80% training and 20% validation sets.

The multi-stage model consists of five neural networks arranged in sequence, each trained on progressively harder and more balanced subsets of the data. At each stage, 40% of the negative samples that the model classifies with the highest confidence are removed, while all positive cases are retained. This iterative filtering gradually increases the effective class ratio from the original $\sim$2% to approximately 15% positive cases. The 40% filter was chosen based on internal testing. Alternative filtering percentages also produced reasonable results, but 40% consistently provided the best trade-off between class balance and overall performance.

All models in the cascade were trained using binary cross-entropy loss and the Adam optimizer with an initial learning rate of 0.001. Training was carried out for up to 50 epochs with a batch size of 32. A ReduceLROnPlateau scheduler (patience of 5, factor 0.5) adjusted the learning rate when the validation loss plateaued, and early stopping with patience of 10 epochs was applied. To improve generalization, weight decay (1e-4) was included in the later stages. All parameters were chosen via testing and comparing the results. Training was conducted on a single Nvidia 3060 Ti GPU, with a fixed random seed to ensure reproducibility.

Stage 1: A ResNet18 architecture with Squeeze-and-Excitation (SE) blocks trains on the full training dataset and filters 40% of the training data based on confidence.

Stage 2: A SimpleCNN is trained with the remaining training subset and filters 40% of the training data again.

Stage 3: An AttentionCNN is trained on approximately 36% of the original healthy training data and all of the Chagas data.

Stage 4: A second SimpleCNN is trained on the remaining subset ($\sim$22%).

Stage 5: A ResNet18 with a Convolutional Block Attention Module (CBAM) trains on the last and most balanced dataset containing roughly 15% of the original training data of which 15% is Chagas data.

The final component of our architecture is an XGBoost meta-ensemble model that combines predictions from all stages to generate the final classification. From each stage, the best model based on validation AUC is saved and then these models predictions serve as the features for the XGBoost. The model is then trained using 5-fold cross-validation on the validation set, where the validation set is repeatedly split into 80% training and 20% testing portions across five folds. The meta-model assigns different weights to each base model according to their predictive importance based on the validation set. This weighted ensemble approach produces the final classification decision.

During testing, samples are processed through all five stages of the pipeline, generating five separate predictions. These predictions serve as input features to the meta-model, which classifies the probability of Chagas disease presence. For each test signal, this process is repeated across five random clips, and the resulting probabilities are averaged to obtain the final prediction.

2.2.2. ResNet18-Based Models

Our implementation adapts the ResNet18 architecture used previously by multiple studies [9–11]. The ResNet architecture in our implementation consists of an initial convolutional layer followed by four residual blocks, each containing two convolutional layers with skip connections. Depending on the stage, the residual blocks are enhanced with either SE blocks [12] or CBAM [13]. CBAM has been proven to work well with both ResNet and ECGs before [14, 15]. The network ends with adaptive average pooling and a fully connected layer for classification.

2.2.3. CNN-Based Models

Our cascade framework incorporates two CNN architectures that complement the ResNet models at different stages of the filtering process.

The SimpleCNN serves as a lightweight architecture in stages 2 and 4 of our pipeline. It consists of three convolutional blocks that progressively extract features while reducing temporal resolution. After feature extraction, an adaptive average pooling layer consolidates the features, followed by a two-layer classifier with 64 hidden units.

For stage 3, we employ AttentionCNN, which enhances feature learning through an integrated attention mechanism. This model follows a similar architectural pattern as the SimpleCNN, but adds an SE layer after the third convolutional block to improve the channel-wise feature responses.

3. Results

We evaluated our cascade architecture using different filter percentages to balance data distribution in later training stages. Among the thresholds tested, the 40% filtering achieved the highest challenge score and consistently strong AUC across stages, indicating the most effective trade-off between data volume and class balance. Higher or lower filtering percentages resulted in slightly lower performance as seen in Table 1.

Filter	AUC Score					
	S1	S1-2	S1-3	S1-4	S1-5	Meta
25%	0.817	0.826	0.836	0.833	0.839	0.844
40%	0.812	0.842	0.856	0.860	0.859	0.864
50%	0.817	0.832	0.839	0.849	0.851	0.856

Table 1. Performance Metrics Across Different Filter Percentages. S1 = Stage 1, S1-S2 = Stages 1 and S2.

Filter	Model Importance				
	S1	S2	S3	S4	S5
25%	5.01	5.22	4.98	4.35	3.14
40%	7.03	5.03	4.36	3.59	2.72
50%	5.31	5.46	4.73	3.46	2.87

Table 2. Model Importances in Meta-ensemble.

In Table 2 we can see the importance of different models in the ensemble. In particular, the early stages consistently contribute more substantially to the ensemble than the later stages, but the later stages also seem to have value. This is demonstrated by the increase of the AUC scores throughout the stages as seen in Table 1.

Challenge score	**Ranking**
0.224	14/41

Table 3. Challenge score for team bug busters, including the ranking of our team on the hidden test set.

The final challenge score in the official phase is lower than the one gathered in our internal tests conducted with the same models. With training dataset, we achieved an average of 0.495 Challenge score with 5-fold cross validation. However, these are not directly comparable.

4. Discussion and Conclusion

Our cascade approach effectively addressed the extreme class imbalance in detecting Chagas disease from ECG recordings. Progressive filtering based on ranking effectively improved the proposed model. Performance metrics show that by filtering 40% of the data, each stage provides an optimal balance between data volume and class distribution. Further optimization of filtering percentages between individual stages represents a potential avenue for refinement. The final result of 0.224 is decent, but we believe that our method was overfit with the training set and thus decreased our result significantly in the final phase.

Our approach has a few obvious limitations. The computational requirements for training five different models are greater than those for training just one. However, having a set of multiple models usually increases performance. Another limitation is that we intentionally reduce the amount of training data. It is generally accepted that more data results in better models, but since the imbalance was so extreme, the benefits of improved class distribution outweighed the costs of a reduced sample size.

Future work could focus on adding data augmentation and using more complex models as part of the ensemble. Data augmentation was briefly tested by us, but our preliminary tests showed overfitting and thus the idea was discarded. This may be due to the low amount of Chagas data and limited representation of ECG manifestations of Chagas disease in the data. Another improvement could be to incorporate more complex models, such as transformers, into the cascading ensemble. We did test multiple different models, but they did not seem to work as well as the combination chosen. Different loss functions were also tested to improve the scores. We implemented various methods to punish the models for false negatives or uncertain positives. However, based on internal testing the classical binary cross-entropy was deemed to work the best.

In conclusion, our cascading methodology demonstrates a promising approach for detecting Chagas disease from ECGs, potentially enabling more efficient screening in endemic regions. The framework's success in handling extreme class imbalance may extend to other rare disease detection tasks in clinical settings. This work also highlights an important real-world consideration: despite achieving good AUC scores, healthcare resource constraints mean each false positive would require expensive serological confirmation, potentially wasting limited resources. This practical constraint emphasizes the value of the challenge's specialized scoring metric.

References

[1] Reyna MA, Koscova Z, Pavlus J, Saghafi S, Weigle J, Elola A, et al. Detection of Chagas Disease from the ECG: The George B. Moody PhysioNet Challenge 2025, 2025. URL https://arxiv.org/abs/2510.02202.

[2] Reyna MA, Koscova Z, Pavlus J, Weigle J, Saghafi S, Gomes P, et al. Detection of Chagas Disease from the ECG: The George B. Moody PhysioNet Challenge 2025. In Computing in Cardiology 2025, volume 52. 2025; 1–4.

[3] Ribeiro A, Ribeiro M, Paixão G, Oliveira D, Gomes P, Canazart J, et al. Automatic diagnosis of the 12-lead ECG using a deep neural network, 2020.

[4] Cardoso C, Sabino E, Oliveira C, de Oliveira L, Ferreira A, Cunha-Neto E, et al. Longitudinal study of patients with chronic chagas cardiomyopathy in brazil (SaMi-Trop project): A cohort profile, 2016.

[5] Wagner P, Strodthoff N, Bousseljot RD, Kreiseler D, Lunze FI, Samek W, et al. PTB-XL, a large publicly available electrocardiography dataset, 2020.

[6] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 2000;101(23):e215–e220.

[7] Sadreddin A, Sadaoui S. Training and testing cascades for imbalanced data classification. In 2022 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2022; 261–268.

[8] Johnson JM, Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data 2019;6(1):1–54.

[9] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016; 770–778.

[10] Zhao Z, Fang H, Relton SD, Yan R, Liu Y, Li Z, et al. Adaptive lead weighted ResNet trained with different duration signals for classifying 12-lead ECGs. In 2020 Computing in Cardiology. IEEE, 2020; 1–4.

[11] Leinonen T, Wong D, Vasankari A, Wahab A, Nadarajah R, Kaisti M, et al. Empirical investigation of multi-source cross-validation in clinical ECG classification. Computers in Biology and Medicine 2024;183:109271.

[12] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018; 7132–7141.

[13] Woo S, Park J, Lee JY, Kweon IS. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV); 3–19.

[14] Luo Y, Wang Z. An improved ResNet algorithm based on CBAM. In 2021 International Conference on Computer Network, Electronic and Automation (ICCNEA). IEEE, 2021; 121–125.

[15] Wang C, Ma J, Wei G, Sun X. Analysis of cardiac arrhythmias based on ResNet-ICBAM-2DCNN dual-channel feature fusion. Sensors 2025;25(3):661.

Address for correspondence:

Jonas Sandelin
Vesilinnantie 3, 20500, Turku, Finland
jojusan@utu.fi