Using a TensorFlow Lite LSTM Deep Learning Model to Screen for Chagas Disease

Zoe Boysen[1], Ron Yang[1], Emilia Fallman[1]

[1]Ephemeris Labs, United States of America

Abstract

As part of the George B. Moody PhysioNet Challenge 2025 under the team name Ephemeris Labs, we developed a deep learning algorithm to detect late-stage Chagas disease in electrocardiogram signals (ECG). Chagas disease, transmitted by the parasite Trypanosoma cruzi, causes almost 10,000 deaths per year but affects 6.5 million people. An efficient diagnosis can help physicians provide an adequate treatment plan, but serological testing is limited. We propose a deep learning model that would test for cardiac alterations associated with Chagas and could be put on a Raspberry Pi, making it more accessible to hospitals. The use of deep learning-based models is a rapidly growing research topic in the medical field for classification and diagnostics. In addition, equipment is not designed for low-cost materials, open source code, and data sets to ensure better accessibility. In this work, we create a long- and short-term memory network to tackle this problem using the TensorFlow library on the George B. Moody PhysioNet Challenge open-source dataset. Under the challenge, our model scored 0.05 (ranked 343 out of 369). In our development outside the official scoring, we present a promising model for live classification on a Raspberry Pi 4 and an evaluation of our TensorFlow Model and TensorFlowLite FlatBuffers to demonstrate their minimal run-time requirements while maintaining an acceptable accuracy of 98%. By focusing on open-source and portable solutions, our approach offers a scalable method to prioritize the testing and treatment of Chagas disease, particularly in resource-limited settings.

1. Introduction

Chagas disease, or American trypanosomiasis, is a parasitic disease caused by the *Trypanosoma cruzi* parasite, spread by *triatomine* (kissing bugs) primarily in Latin America. Acute chagas can be treated in the early stage but is often asymptomatic and therefore develops into a chronic disease. Those with late stage chronic Chagas disease will develop often some type of cardiomyopathy, which can lead to heart failure and increase likelihood of death.

The George B. Moody 2025 Challenge invites teams to develop automated, open-source algorithms for identifying cases of Chagas disease from electrocardiograms (ECG) [1, 2]. Accurate detection of arrhythmia abnormalities is key to diagnosis. An efficient diagnosis can help physicians provide an adequate treatment plan, but serological testing is limited. Electrocardiograms (ECGs) remain a gold standard for measuring heartbeats and diagnosing arrhythmia. They are applicable in assessing the severity of a disease, monitoring patients in clinical trials, and screening individuals for high-risk occupations [3]. Developing a low-cost, efficient device will help with the growing need of diagnosing cardiac arrhythmia and potentially save many lives.

Typical ECG tests usually span in the length of minutes. If a signal is inconclusive, the patient may need to wear a Holter monitor for all-day heart monitoring [4]. One study was able to integrate an AI model onto a Holter monitor for 24-hour monitoring to detect atrial fibrillation [5]. The major problem of this study was their use of Holter data for the training of their model. The dimensions of this data are magnitudes larger than standard ECG data, requiring much more training and data to achieve an optimal model. An AI-enabled ECG test helps physicians reduce the number of inconclusive tests and is more practical than an AI-enabled Holter monitor. This drives the need for further research into deep learning embedded devices for ECG classification. Most technologies in this realm are inefficient for resource-constrained IoT devices and embedded systems and require raw data transmitted to a remote server [6]. A simpler and low-cost solution must be examined more deeply.

Deep learning (DL) is a new and evolving technology being applied in the classification of medical data for prevention and diagnostics. DL is part of machine learning that models patterns similar to the structure of the brain. An example of this kind of data processing and prediction is through ECG arrhythmia classification.

Both Raspberry Pi and Arduino are low-resource and cost circuits that support the use of TensorFlow Lite mod-

els. The Raspberry Pi is a single-board and modular computer meant for high-performance, low-cost, general-purpose computing platforms built on the ARM architecture and running the Linux operating system [7]. We see this as an opportunity to develop methods for high access, low-cost, healthcare.

In the following sections, we highlight the development of an LSTM deep-learning framework to classify Chagas disease in a patient dataset. With further development, the TensorFlow Lite model could be an important tool in arrhythmia detection and diagnosis onto an embedded system. Our work aims to build on the work done by others and introduce new potential areas of research in this field before a low-cost, effective embedded system can be developed for clinical settings.

2. Methods

The goal of this study is to present a working model and workflow of an ECG arrhythmia classification on a Raspberry Pi and Arduino system. We present a TensorFlow and TensorFLowLite LSTM models able to label if a patient's ECG signal shows signs of Chagas disease or not from the provided datasets of the challenge[8–12]. Each data set contains a padded raw 12-lead ECG single (DI, DII, DIII, AVR, AVL, AVF, V1, V2, V3, V4, V5, V6) with a sampling frequency of 400 Hz and duration of 10s/7s. Our method demonstrates arrythmia detection through ECG classification. Furthermore,our model can be converted to a form used on microcontroller systems, such as the Raspberry Pi Pico, for future implementations of a low-cost device capable of flagging irregular signals. All materials and methods are easy to follow and are open to the public through GitHub.

For the model's architecture, TensorFlow's LSTM method was chosen for its benefits in time series forecasting and anomaly detection [13]. LSTMs can learn and use time series data to make predictions or flag the data found to be an anomaly.

Long short-term memory (LSTM) networks is a predominantly used to learn, process, and classify sequential data because these networks can learn long-term dependencies between time steps of data [14, 15]. The LSTM network is a recurrent neural network (RNN) that deals with the vanishing gradient problem present in traditional RNNs through a gating mechanism controlling the flow of information across the network [14, 15]. This makes the model great for machine translation, speech recognition, natural language processing, video analysis, and time series forecasting. [15].

It is important to note that to ensure the best output and the least amount of errors, our model has the following library dependencies in python:

- joblib==1.4.2
- numpy==2.0.2
- pandas==2.2.2
- scikit-learn==1.6.0
- wfdb==4.1.2
- h5py==3.12.1
- tensorflow==2.19
- keras==3.9.0
- PyWavelets==1.8.0
- scipy==1.15.2

These are listed in our requirements text file, and can be used to set up a Docker image or virtual machine to match the environment we run in.

2.1. Feature Extraction

The features used in the model are the sex and approximate age of the patient, and a raw ECG 12-lead signal. The sex is represented by a one-hot encoded 1 by 3 matrix of whether the patient is female, male, or other. Since raw ECG signals often have missing data or corrupted leads, we check for finite signals and perform some signal processing before passing them into the model. On each lead, the mean and standard deviation are taken as a form of processing to help with training speed and detection of patterns. It is a way to denoise and bring out key differences in signals.

2.2. Model Architecture

Our model contains 3 LSTM layers and one dense layer, shown in Figure 1.

The LSTM layers run with a batch size of 16, a number divisible by the training, validation, and testing data set lengths. They also have an input shape of a maximum data chunk length of 1 and a dropout of 0.2. Each of these layers returns a set of vectors with a dimension of the batch size. The final layer of the model is a dense layer that fully connects the previous layers and changes the output dimension to match the number of units specified [15]. As shown in Figure 1, the dimensions of each layer are present. This illustrates that the input shape is 3-dimensional and the final output is a 2-dimensional representation of if the signal indicates the presence of Chagas disease.

The input shape is given by batch size (16), the time step (7), and the input dimension, number of features (1). The time step was determined by the features extracted ([age], one-hot encoding sex, [signal mean, signal standard deviation]) and is the length of the features. These values influence the sizes of the subsequent layers, where the final layer outputs a shape of batch size and number of labels (2).

Within each LSTM layer, there are 192 LSTM cells- the value of time steps (t)- that run the specific functions and matrix math of the model. This is exhibited in Figure 1.

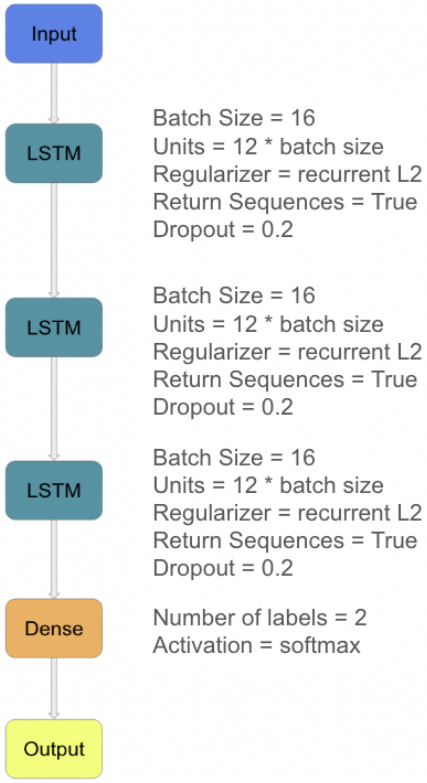

Figure 1. LSTM Model Architecture for the detection of Chagas disease

We used the standard sigmoid functions and tanh activation functions as they are the most commonly used. To capture the full complexity of the dataset, the units value was set to 192. This increases the number of neurons at each gate's functions within each LSTM cell. Furthermore, this increases the number of parameters at each layer, making the overall model more complex. All cells build on each other by passing the cell state and hidden state to the next in the series.

The model ran with L2 recurrent regularization of value 0.01 to help with overfitting. In testing, we saw that adding the typical ReLU activation layer caused an issue of the loss equaling infinity- exploding gradient problem, so we stuck with the tanh function. The model ran for 10 epochs for training. Upon compilation, the sparse categorical cross-entropy loss function and Adam optimizer were used. Both are typical methods for LSTM networks, but this particular loss function helps with numerical and multiple matching.

2.3. Model Output

The model output comes from the final layer in our LSTM model architecture. We used a dense layer with softmax activation in order to obtain a binary output with the predicted probability of each label.

2.4. TensorFlow Lite

Our model, is built as TensorFlow but for future applications we convert it to a TensorFlow lite for model size reduction. The compression of the model is pertinent for an embedded system, model performance over a live ECG signal, and faster computations on the CPU.

3. Results

In this section, we present the results of our model for the detection of Chagas disease. Using LSTM, we were able to place in 342nd place in the George B. Moody Challenge. We received a score of 0.05, where the score represents the true positive rate out of the total number of Chagas cases. In addition to this score, our model is able to obtain about 98% training accuracy with a loss of about 0.10 after 10 epochs.

Training	Validation	Test	Ranking
0	0.05	-	342/369

Table 1. Challenge scores for our selected entry (Ephemeris Labs), inclusion of the test set will be done after the conference. We use a LSTM model with the given dataset to detect for Chagas disease.

4. Discussion and Conclusions

In this paper, an LSTM-based classification algorithm was proposed to label a 12-lead ECG signal as having Chagas disease or not. This model has a training accuracy of about 98%. There is potential capability to run live or recorded signals on a Raspberry Pi and Arduino device. The goal of this work was an easy/versatile way to train and transport a working model onto a Raspberry Pi and Arduino system. In this process, we wanted to thoroughly test the model for its use on a low-cost, simple computing device. The end program could take in any ECG signal and flag abnormalities on a wearable device. This device would be accessible to most and easy to use, a great advantage for the medical field.

We were able to achieve a loss of 0.1 and an accuracy of 98%. Likely, the 98% training accuracy is due to the imbalance of positive Chagas cases with the number of negative Chagas cases. Having this imbalance will inflate the accuracy as the model will try and lean towards the label with the larger number of samples. Like other works, we found that LSTM is a great model for classifying ECG signals. We also concluded that more LSTM layers and adjustments in the hyper-parameters provided better outcomes.

To improve on this model, we believe changing the input features would be key in the successful detection of Chagas disease. As LSTM has proven to be effective in handling signals well, we expect to see the successful implementation of an LSTM model on the ECG signals themselves as a successful method to detect Chagas. This could be achieved through signal processing over the 12 leads. This includes padding to the largest signal, down-sampling by a factor of 4, performing a sym 3 wavelet transfer and z-score normalization. One key issue, when developing this model is the memory allocation needed to successfully train this type of model. In future work, this will need to be addressed to train on the entirety of a 12-lead ECG dataset.

Acknowledgments

We declare there is no conflicts of interests. We would like to thank all the mentors within this competition journey that has allowed for our success.

References

[1] 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.

[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. Computing in Cardiology 2025;52:1–4.

[3] Sattar Y, Chhabra L. Electrocardiogram. In StatPearls [Internet]. StatPearls Publishing, 2023; .

[4] professional CCm. Holter monitor: What it is, uses, test details results. URL https://my.clevelandclinic.org/health/diagnostics/21491-holter-monitor.

[5] Kim JY, Kim KG, Tae Y, Chang M, Park SJ, Park KM, et al. An artificial intelligence algorithm with 24-h holter monitoring for the identification of occult atrial fibrillation during sinus rhythm. Frontiers in Cardiovascular Medicine 2022;9:906780.

[6] Sharma K, Eskicioglu R. Deep learning-based ecg classification on raspberry pi using a tensorflowlite model based on ptb-xl dataset. arXiv preprint arXiv220900989 2022;.

[7] Pi R. URL https://www.raspberrypi.com/about/.

[8] 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. Nature Communications 2020; 11(1):1760.

[9] 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. BMJ Open 2016;6(5):e0011181.

[10] Wagner P, Strodthoff N, Bousseljot RD, Kreiseler D, Lunze FI, Samek W, et al. PTB-XL, a large publicly available electrocardiography dataset. Scientific Data 2020;7:154.

[11] Nunes M, Buss L, Silva J, Martins L, Oliveira C, Cardoso CS BB, et al. Incidence and predictors of progression to chagas cardiomyopathy: Long-term follow-up of trypanosoma cruzi-seropositive individuals. Circulation 2021; 144(19):1553–1566.

[12] Pinto-Filho M, Brant L, Dos Reis R, Giatti L, Duncan B, Lotufo P, et al. Prognostic value of electrocardiographic abnormalities in adults from the brazilian longitudinal study of adults' health. Heart 2021;107(19):1560–1566.

[13] Team K. Keras documentation: Lstm layer. URL https://keras.io/api/layers/recurrent_layers/lstm/.

[14] Hung CL. Chapter 11 - deep learning in biomedical informatics. In Zheng Y, Wu Z (eds.), Intelligent Nanotechnology, Materials Today. Elsevier. ISBN 978-0-323-85796-3, 2023; 307–329. URL https://www.sciencedirect.com/science/article/pii/B9780323857963000111.

[15] GeeksforGeeks. Deep learning: Introduction to long short term memory, 12 2023. URL https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/.

Address for correspondence:

Zoe Boysen
231 Park Drive, Boston, Massachusetts, 02215, United States of America
zoe.elizabeth.boysen@gmail.com

Ronald Yang
2034 Kickapoo Trail, Vineland, 08361, New Jersey, United States of America
yangronald27@gmail.com