Robust Classification of Cardiac Arrhythmia Using a Deep Neural Network

Connor Lennox[1] and MD Shaad Mahmud[2]

***Abstract*— Machine learning has become increasingly useful in various medical applications. One such case is the automatic categorization of ECG voltage data. A method of categorization is proposed that works in real time to provide fast and accurate classifications of heart beats. This proposed method uses machine learning principles to allow for results to be determined based on a training dataset. The goal of this project is to develop a method of automatically classifying heartbeats that can be done on a low level and run on portable hardware.**

I. Introduction

Real time categorization of ECG heart beats allows for a continuous, accurate monitoring system. Due to the sporadic nature of some heart conditions, this type of careful observation is required, since any non-automatic method of detection is not feasible on a long-term basis. Additionally, in recent years machine learning has become more and more accurate in the field of data categorization. In particular, the Python package of TensorFlow allows for this style of categorization, along with real-time capabilities with a pre-trained learning model. A method of preprocessing data for this form of analysis is presented and allows for a consistent style of data input, which is a critical component for accurate machine learning results. Preprocessed data is able to be generated in real-time, and so all readings put into the categorical model for analysis are of the same format. Experimental results with testing done on the MIT-BIH Arrhythmia Database suggest sufficient accuracy for categorization of different types of heart beats [1]. After creation of this categorical model, evaluations were done to ensure its accuracy and effectiveness in beat classification. Additionally, in order for this technology to be used for its main intent of providing real-time monitoring for people at risk of irregular heart beats, there must be some method of implementation. A proposed method of implementation includes the usage of the Apollo 3 Edge (Ambiq Micro.).

II. Methods

The proposed method of heart beat analysis is designed to offer real-time beat categorization and as such is built around an incoming stream of data. Python was chosen as a platform for the implementation of the method, as to promote portability and the use of other code packages that would serve useful throughout the process. As this is a machine-learning based approach, the first step in creating a functional model is to train the model against data with known categorizations. [6].

The Sparkfun Edge Apollo 3 is a low-power microcontroller board designed specifically for long battery life and portable applications. It features a high-performance ARM Cortex M4 processor, and an integrated low-power Bluetooth module. The prototyping board is developed by Sparkfun Electronics, and production was started in March 2019. This microcontroller is, as of writing, a cutting edge component in small-scale machine learning, as it has the small size to allow usage in many different applications, alongside the processing power to support potential use cases. In addition to these state-of-the-art components, the Apollo 3 is also designed to run a slimmer version of TensorFlow, called TensorFlow Lite. This version of TensorFlow is fully capable of making predictions given input data, however it is not capable of creating its own models for use in making predictions. [8]

The MIT-BIH Arrhythmia Database is a set of readings taken at labs owned by Boston's Beth Israel Hospital (now known as the Beth Israel Deaconess Medical Center) and MIT. The dataset contains 48 half-hour samples of ECG recordings, obtained from 47 subjects between 1975 and 1979. Since its release in 1980 it has become a standard for heartbeat classification and serves as a strong foundation to build the TensorFlow model off of. The samples found in the MIT-BIH Arrhythmia Database are digitized recordings of ECG data, which was originally written on 9-track half-inch digital tape at 800 and 1600 bpi. The recordings were digitized at 360 samples per second per channel, with 11-bit resolution over a 10 mV range. For each record, at least two cardiologists categorized each beat, and the final decided upon classifications for any given beat in the dataset is coupled with the data [1]. The different types of beat, alongside the corresponding character codes, can be found in Table 1.

A. Data Preprocessing

An important aspect of the categorical model is consistent, regular data. [7] In order to ascertain that the data received by the model is in a similar form regardless of the input, a process was created to regulate the data before categorization. The steps of this process are as follows:

- Threshold Calculation: In order to properly detect peaks, a threshold value needs to be determined. This threshold represents the minimum value that a data point

[1] Connor Lennox is with the Department of Computer Science from the University of New Hampshire.

[2] MD Shaad Mahmud is with the Department of Electrical and Computer Engineering from the University of New Hampshire.

TABLE I

Beat types in the MIT-BIH Arrhythmia Database

Beat Type	MIT-BIH Annotation	Categorical Model Index
Normal	. or N	0
Left bundle branch block beat	L	1
Right bundle branch block beat	R	2
Bundle branch block beat	B	3
Atrial premature beat	A	4
Aberrated atrial premature beat	a	5
Nodal premature beat	J	6
Supraventricular premature beat	S	7
Premature ventricular contraction	V	8
R-on-T premature ventricular contraction	r	9
Fusion of ventricular/normal beat	F	10
Atrial escape beat	e	11
Nodal escape beat	j	12
Supraventricular escape beat	n	13
Ventricular escape beat	E	14
Paced beat	/	15
Fusion of paced/normal beat	f	16
Unclassifiable beat	Q	17
Beat not classified during training	?	18

can have in order to be recognized as a "beat". The threshold of a given window of data points is calculated as the square root of the RMS of the data. This value provides a conservative, yet effective baseline for peak detection.

- Peak Detection: Peaks are determined to be data points that meet two criteria:
  - The value must be greater than or equal to the calculated threshold value
  - The value must exist at a zero-crossing of the first derivative of the data.

  In order to perform peak detection, the first derivative of the data is calculated. Then, the zero-crossings of this derivative are compared to the previously calculated threshold value. When a value is found to meet both criteria, it is labelled as a beat.
- Beat Extraction: The amount of data points that are collected to represent a beat are determined by the mean peak-delta. This value is calculated as data is observed and represents the average amount of data points between two detected beats.
- Normalization: Data is scaled by a factor of max(data) – min(data), to ensure that the data points themselves all fit within a range of 1. A transformation is then applied to the data to bring it into the standardized range of (0, 1). This transformation consists of a vertical shift by min(data). It is of note that this minimum is representative of the minimum after the scaling.
- Padding: Beats that do not have 300 samples are padded with zeros so that all beats fed into the model are of the same length. It should be noted that beats that consist of more than 300 samples are truncated to fit within the target window.

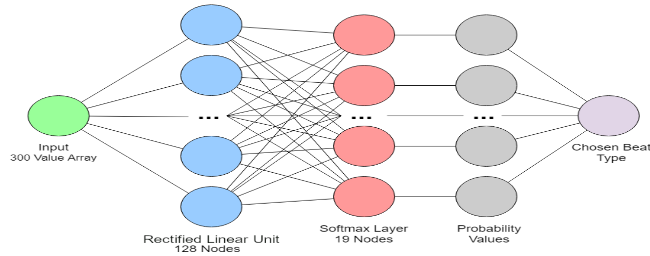

Fig. 1. An image depicting the proposed structure of the machine learning model used for categorization.

B. Model Design

The next step after handling the data was to define a structure for the learning model. To aid in the ease of use of the model, a very simple design was created, which would still provide adequate results. The clear start to this type of model would be a single input, as even though 300 values are being passed to the model, they are passed in as a single stream. After this input layer, a 128-node dense layer was created. This layer handles the main calculations of the categorization process, and so it was important that enough nodes were present to allow proper pattern detection during the learning process. A rectified linear unit activation function was chosen for this layer, as it provides effective learning processes and results in accurate heart beat categorization [4].

The final layer in the learning model is a 19-node dense softmax layer. In the context of this model, the value in a given node of this layer is the confidence the model has in the provided beat being of any given type. In most cases, the beat with the highest confidence is what the beat will be classified as by the model. Later a method of ignoring classifications with low confidence values will be introduced, allowing higher overall accuracy at the cost of possibly ignoring beats that would have been correctly classified.

III. Evaluation

Initial analysis of the categorical model has led to a 99.256% accuracy over training data, and a 96.044% accuracy over testing data. Accuracies were calculated based on percentage of beats categorized correctly. The training data set contained 101560 beats, leading to a count of approximately 100804 beats correctly categorized. The testing data set contained 5814 beats, leading to a count of approximately 5584 beats correctly categorized. These results match other current methods of heart beat classification. [7]

The time taken for any given classification is negligible compared to the time between heart beats, and so the proposed method of beat recognition and categorization can be considered to be truly "real-time". That is, a given beat can be classified before the next occurs. Because beats are only sent for classification when they reach the center of the active window, the duration of time from when a beat occurs to when it is considered is determined by the active window. For most testing purposes, an active window of 3

TABLE II

RESULTS OF MODEL EVALUATION

Dataset	Number of Beats	Beats Correctly Classified	Accuracy
Training Data	101560	100804	99.256%
Testing Data	5814	5584	96.044%

seconds was used, which allowed for enough data to properly calculate threshold values, while also not having so much data that there was too much time between when a beat occurred and when it was sent for classification. With the 3 second active window in use, there is a 1.5 second delay between the peak of a beat occurring and it reaching the center of the active window for classification. However, this is somewhat alleviated by the fact that a beat is not over once the peak occurs - many critical structures are located after this part of the wave. As such, there is only about a half second delay between the beat fully occurring and its evaluation via the machine learning model.

Although the model may not always be correct, there are ways to mitigate unwanted consequences from an incorrect categorization. Because of the softmax layer in the learning model, each of the possible beat types for classification are given a value that represents how sure the model is that the beat is of that type. With most correct predictions, this value is very close to 1, taking up most of the available space within the softmax layer [12]- [13]. This is due to the softmax property that the values of all the nodes must sum to 1. When the model makes an incorrect prediction, it is often not as sure about its answer. By looking at how certain the model is on a given prediction, preemptive measures can be taken to ensure that only correct categorizations are being used to make decisions regarding future actions.

By varying the minimum confidence level required in order for a concrete decision to be made, the accuracy of the model can be increased [14]. However, this also comes at a risk of rejecting correct categorizations that did not have a high enough confidence level. Further analysis of this method is done in the Results section of this paper, and serves to show the direct effect of this technique on the accuracy of the model. Further relative speed-ups between the occurrence of a beat and its classification could be achieved by considering a beat after a calculated amount of time, equal to the mean RR interval divided by two. This method was not used in testing as it had the potential to cut off critical data points on the tail end of a beat. It was decided that having a slightly slower response time in regard to irregular beats was preferable to having less accurate categorizations.

IV. RESULTS

The proposed implementation of this trained categorical model is through the Sparkfun Edge Apollo 3 microcontroller board or an equivalent. Due to the board's ability to run TensorFlow Lite, it is a prime candidate for the implementation of this model. Since the model has already been created and trained, there is no need for the final implementation to modify it. Since this is the main

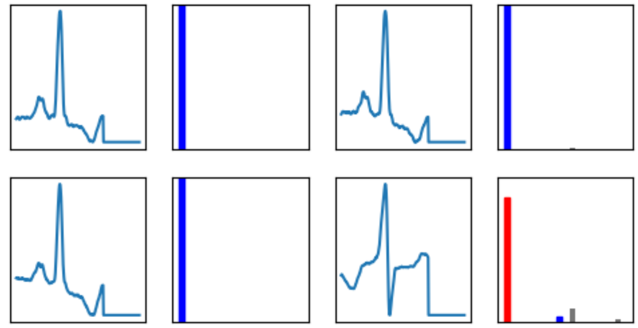

Fig. 2. A grid of beats and their corresponding prediction list. Each bar represents how sure the model was of each possible prediction for the given beat. In each plot, the blue bar represents the correct answer, while the one red bar represents an incorrect answer given by the model. This plot shows that the model is usually quite certain about the answer it provides, even when that answer may be incorrect. However, there is still some doubt in the models answer when it gave the incorrect answer, as it only had a certainty value of 85.76%.

feature that TensorFlow Lite lacks over the full version of TensorFlow, there is effectively no difference in the end results. [8]

Further analyses have been done on the categorical model, with additional care to filter out classifications that may be incorrect through the usage of a confidence threshold. The first analysis done was in regards to model accuracy in relation to a minimum confidence value, used as a threshold to ignore classifications that the model is sufficiently unsure about. In this study, several variables were tracked. First, the total number of classifications that were considered to be valid was tracked, in order to better understand how the scale of the amount of beats that had been discarded. Secondly, the accuracy within the set of valid classifications was calculated. By looking at these two values, it can be seen that by increasing the minimum confidence value, the overall accuracy is increased, with the trade-off being the small percentage of beats which are ignored. The results in the following table come from the testing data derived from the MIT-BIH dataset, with additional filtering to discard beats that have a classification confidence below the given threshold values.

TABLE III

EFFECT OF CONFIDENCE THRESHOLD ON MODEL ACCURACY

Confidence	Accepted Beats	Accuracy	Rejected Beats	% Accepted
0.00	5814	96.03	0	100.00
0.80	5785	96.35	29	99.50
0.85	5775	96.42	39	99.33
0.90	5756	96.61	58	99.00
0.95	5718	96.89	96	98.35
0.99	5562	97.68	252	95.67

From these results, it can be seen that by taking into consideration a minimum confidence value (and subsequently increasing that value) the effective accuracy of the model can be increased. However, this comes at the drawback of losing up to 4.33% of classifications. Even so, this is a relatively safe operation, as out of the 252 beats discarded with a 0.99 minimum confidence, 150 were beats that had previously been incorrect classifications. This corresponds to a 1.83% decrease in the total number of correct classifications, and a 64.94% decrease in the total number of incorrect classifications. It is safe to conclude that this small dip in correct classifications is justifiable given the large benefit in the removal of such a large portion of the incorrect classifications.

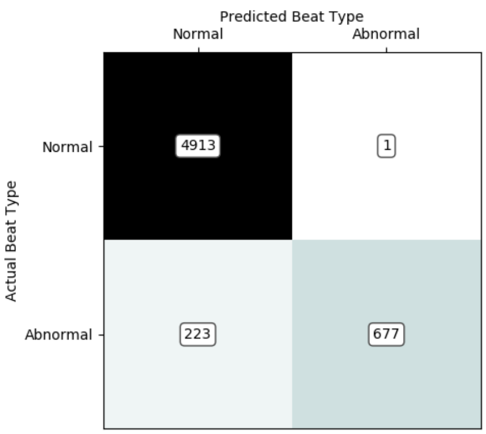

Fig. 3. A confusion matrix showing the accuracy of the model when considering only whether or not a beat is abnormal or normal.

It should also be noted that the MIT-BIH Arrhythmia Database includes specifications for three beat types that are not present in the dataset itself. These three beats are types 3, 9, and 13, representing bundle branch blocks, R-on-T premature ventricular contractions, and supra-ventricular escape beats [11]. The reason for this disconnect in specified beats and beats present in the data is because these three extra beat types exist in the PhysioBank annotation set, however they were not used within the MIT-BIH dataset. Because the MIT-BIH dataset is available through PhysioBank, the labels have been adjusted to match the standardized annotations.

The usage of a minimum confidence value has been shown to have a considerable effect on the accuracy of the model. Although some correctly categorized beats are lost due to a low confidence, the ratio of discarded incorrect beats to discarded correct beats is large enough to justify this loss in most scenarios. There must be some consideration for scenarios in which every abnormality must be caught without fail, in which the entire confidence level threshold can be skipped in favor of taking every classification into consideration.

V. Conclusion

A method for real-time heart beat detection and categorization has been presented. This machine-learning based model allows for consistent, accurate, but most importantly fast categorizations of heart beats from ECG data. Normalized, succinct input data further increases this accuracy and prevents common machine learning problems, specifically the problem of over-fitting. Additionally, a parallel method of preprocessing ECG data is presented which works in conjunction with the categorization model. This preprocessing method allows for data to be rewritten in a format that mirrors the data that the categorization model was trained with, allowing for more accurate categorization to occur. Analyses were presented that show the accuracy of the model when categorizing over two potential use cases-the categorization of a specific type of abnormality and the simple detection of present abnormalities.

References

[1] "MIT-BIH Arrhythmia Database Directory", Physionet.org, 2019. [Online]. Available: https://physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm.

[2] Q. Zhang, D. Zhou and X. Zeng, "Highly wearable cuff-less blood pressure and heart rate monitoring with single-arm electrocardiogram and photoplethysmogram signals", *BioMedical Engineering OnLine, vol. 16, no. 1, 2017*. Available: 10.1186/s12938-017-0317-z.

[3] S. Mahmud *et al.* "SensoRing: An Integrated Wearable System for Continuous Measurement of Physiological Biomarkers", Presented at the *2018 IEEE International Conference on Communications (ICC),* MO, USA, 2018.

[4] S. Mahmud, H. Fang, H. Wang, S. Carreiro, E. Boyer, "Automatic Detection of Opioid Intake Using Wearable Biosensor", *IEEE International Conference on Computing, Networking and Communications (ICNC)*, Maui, Hawaii, 2018.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., *Tensorflow: large-scale machine learning on heterogeneous distributed systems 2016, pp. 1–19 arXiv preprint arXiv: 1603.04467*.

[6] M. Kachuee, S. Fazeli and M. Sarrafzadeh, "ECG Heartbeat Classification: A Deep Transferable Representation", *CoRR*, vol. 180500794, 2018. Available: http://arxiv.org/abs/1805.00794.

[7] G. Sannino and G. De Pietro, "A deep learning approach for ECG-based heartbeat classification for arrhythmia detection", *Future Generation Computer Systems*, vol. 86, pp. 446-455, 2018. Available: 10.1016/j.future.2018.03.057.

[8] A. Fandango, Mastering TensorFlow 1.x. [S.l.]: Packt Publishing, 2018.

[9] S. Hanyu and C. Xiaohui, "Motion artifact detection and reduction in PPG signals based on statistics analysis," *2017 29th Chinese Control And Decision Conference (CCDC)*, Chongqing, 2017, pp. 3114-3119.

[10] A. H. Gee, R. Barbieri, D. Paydarfar, and P. Indic, "Predicting Bradycardia in Preterm Infants Using Point Process Analysis of Heart Rate," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2300-2308, 2017.

[11] A. Verma, S. Cabrera, A. Mayorga and H. Nazeran, "A Robust Algorithm for Derivation of Heart Rate Variability Spectra from ECG and PPG Signals," *2013 29th Southern Biomedical Engineering Conference*, Miami, FL, 2013, pp. 35-36.

[12] S. Savalia and V. Emamian, "Cardiac Arrhythmia Classification by Multi-Layer Perceptron and Convolution Neural Networks," *Bioengineering*, vol. 5, no. 2, p. 35, Apr. 2018.

[13] D. Zafeiris, S. Rutella, and G. R. Ball, "An Artificial Neural Network Integrated Pipeline for Biomarker Discovery Using Alzheimers Disease as a Case Study," *Computational and Structural Biotechnology Journal*, vol. 16, pp. 77–87, 2018.

[14] S. Parvaneh, J. Rubin, S. Babaeizadeh, and M. Xu-Wilson, "Cardiac arrhythmia detection using deep learning: A review," *Journal of Electrocardiology*, 2019.