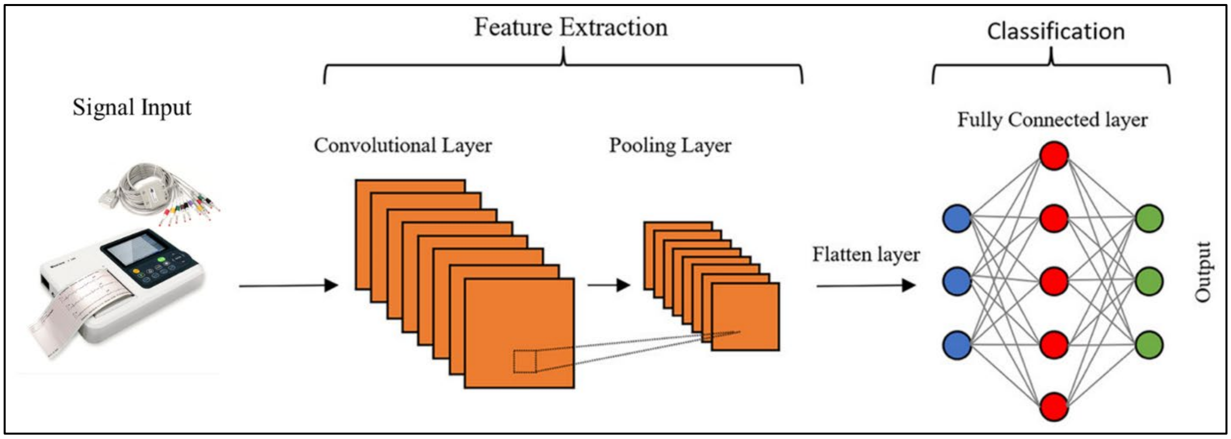
Cardiovascular Disease Detection of a 12-Lead Electrocardiogram Using Machine Learning

Team #01

Team Members: Anna Chau, Luke Dewees, Abraham Paroya

Advisor: Dr. Joseph Picone

Coordinator: Dr. Maryam Alibeik



Senior Design Final Document

Temple University, College of Engineering

April 29th, 2024

Acknowledgements

We would like to extend our sincere gratitude to Dr. Joseph Picone for his invaluable guidance and support throughout this project. His expertise and encouragement has been instrumental in shaping our research endeavors.

Abstract

Heart disease stands as the foremost cause of mortality worldwide, particularly in developing countries where access to cardiologists remains scarce. While low-cost electrocardiogram (ECG) machines are accessible in these areas, these machines lack the capacity to comprehensively assess health factors of the patient such as age, weight, and medical history. In pursuit of an automated diagnostic approach for six potential cardiovascular diseases, a Convolutional Neural Network (CNN) underwent training on a dataset comprising 250,000 electrocardiogram (ECG) examinations supplied by the Telehealth Network of Minas Gerais. The performance of this neural network was compared to a published Deep Neural Network (DNN), as referenced in *"Automatic diagnosis of the 12-lead ECG using a deep neural network,"* by Antônio H. Ribeiro et. Al., and diagnoses from medical students and residents, employing the F1 score metric. The results indicate the CNN surpassed the diagnostic abilities of medical students and residents but fell short of the performance of the DNN detailed in the referenced study.

Table Of Contents

Section 1: Problem Statement

Within developing countries, there is a pressing issue of inadequate access to cardiovascular healthcare, primarily due to the severe shortage of cardiologists. For instance, in Brazil in 2022, there were only 8.40 cardiologists for every 100,000 individuals, while in Africa, a continent with a population of approximately 1.2 billion people, there are only about 2,000 recorded cardiologists. This alarming scarcity of official cardiologists requires the urgent need for innovative solutions, and although low-cost electrocardiogram (ECG) machines are available in these regions, they fall short in providing comprehensive diagnoses due to their reliance on classical and naïve algorithms. These machines often lack the capacity to comprehend a wider range of present cardiovascular disease indictors, reducing their effectiveness in detecting indicators of cardiovascular diseases.

To address this gap, our team proposes the development of a supervised machine learning model tailored for cardiovascular disease detection – specifically of the following diseases: 1st Degree Atrioventricular Block (1dAVb), Right Bundle Branch Block (RBBB), Left Bundle Branch Block (LBBB), Sinus Bradycardia (SB), Atrial Fibrillation (AF), and Sinus Tachycardia (ST). Our model will analyze a diverse range of signal variations to emulate the diagnostic capabilities of experienced medical professionals. This approach allows us to proceed in the correct direction to enhance diagnostic accuracy and expand access to cardiovascular healthcare in resource-constrained environments.

Furthermore, our project aligns closely with several United Nations Sustainable Development Goals (SDG) that emphasize the importance of social responsibility and ethical conduct. Specifically, by striving to improve access to cardiovascular healthcare in developing countries, our project directly contributes to SDG No. 3: Good Health and Well-Being as our team is aiming to reduce healthcare inequalities and promote a healthier well-being for individuals in need. (https://sdgs.un.org/goals, 2024). Moreover, our work also aligns with SDG No. 10: Reduce inequality within and among countries. (https://sdgs.un.org/goals, 2024). As we work towards creating equal access to cardiovascular healthcare, we directly accomplish this specific SDG goal. By contributing to reducing these disparities and promoting healthcare equity, our project highlights our commitment to social responsibility and ethical practices in software.

Section 2: Design Criteria

The table below outlines the design criteria for this project. The primary goal is to replicate the performance metrics outlined in *"Automatic diagnosis of the 12-lead ECG using a deep neural network,"* by Antônio H. Ribeiro et al. Performance metrics derived from an evaluation dataset are provided for a deep neural network system, two 4th year cardiologist residents, two 3rd year emergency residents, and two 5th year medical students. For the students and residents, each student or resident was responsible for the annotation of one half of the evaluation dataset. This project's design criteria hinge on comparing the performance of our team's model to the performance from the previously mentioned published paper's model.

Criteria	Negotiable/Non-Negotiable
Exceed performance of study's Deep Neural Network	Negotiable
Exceed performance of 4th year cardiologist residents	Non-Negotiable
Exceed performance of 3rd year emergency residents	Non-Negotiable
Exceed performance of 5th year medical students	Non-Negotiable
Packaged with compact and easy to use software	Non-Negotiable
Compatible with standard ECG file formats (PHYSIONET, WFDB)	Non-Negotiable

Table 2.1: Design Criteria

In terms of comparing our trained classifier's performance to the metrics published in the chosen paper, exceeding the performance of the 4th year cardiologist residents, 3rd year emergency residents, and the 5th year medical students is labeled as non-negotiable. This is because the goal of the project is to produce a classifier that acts as a cardiologist would, which requires outperforming the results of residents and students. Exceeding the performance of the study's published neural network was labeled as negotiable, as it is our belief that the neural network's training was optimized for performance over a specific evaluation dataset, and matching the results of the study's neural network would be unrealistic.

There are also two additional non-negotiable design criteria. The first is that our software must be packaged with compact and easy to use software. This will require the development of Python tool scripts for data processing, training, and decoding, along with README documents to accompany these scripts. This is non-negotiable because if our software is to be used by other groups, well-documented and packaged code will make sure no time is wasted attempting to set-up our software.

The second additional non-negotiable component of our design criteria is that our software must be compatible with standard ECG file formats, which include PHYSIONET and WFDB file formats. This will be implemented by ensuring the software we use to read ECG exams is compatible with both file formats, and that the ECG data will be converted to a consistent format, regardless of its source file formatting. This is non-negotiable because our goal for the future work of this project is to make it possible to implement our software in hospitals worldwide, where the source file formatting of the ECG exams can vary.

Section 3: Potential Solutions

In the field of machine learning classifiers and algorithms, there are numerous approaches available to pursue, where each one offers its own methodology of complexity. Some machine learning models can be developed using a simple classifier, while some are developed using more complex classifiers, such as deep neural networks. In our project, we opted to explore both non-neural network and neural network systems to conduct a comparative preliminary analysis, before ultimately choosing one system to fully pursue for our machine learning model after determining each systems' performance in terms of their F1 score.

3.1: Random Forest Classifier

The Random Forest classifier (RNF) is a versatile machine learning algorithm known for its effectiveness in classification tasks. It operates by constructing multiple decision trees during its training phase, where each decision tree of the forest is built by using a random subset of the features and training data. During classification, input data is passed through each of the decision trees, and the final classification is determined by a majority vote or averaging of the individual tree predictions. Random Forest classifiers provide insight into the feature importance of the data, which makes them valuable tools for classification tasks across diverse application domains, as they provide informed decision-making and predictive accuracy. To ensure the relevance of the RNF in the context of this study, it is crucial to preprocess the data in a manner that standardizes all Electrocardiograms (ECGs). This involves converting each ECG from its raw 8 leads to the complete set of 12 leads. Subsequently, the signals were flattened, arranging all twelve leads of the ECG signal sequentially within each row of the training table. Each row then had had a corresponding annotation which is a .csv file containing 6 columns representing the presence or absence of a disease in that exam.

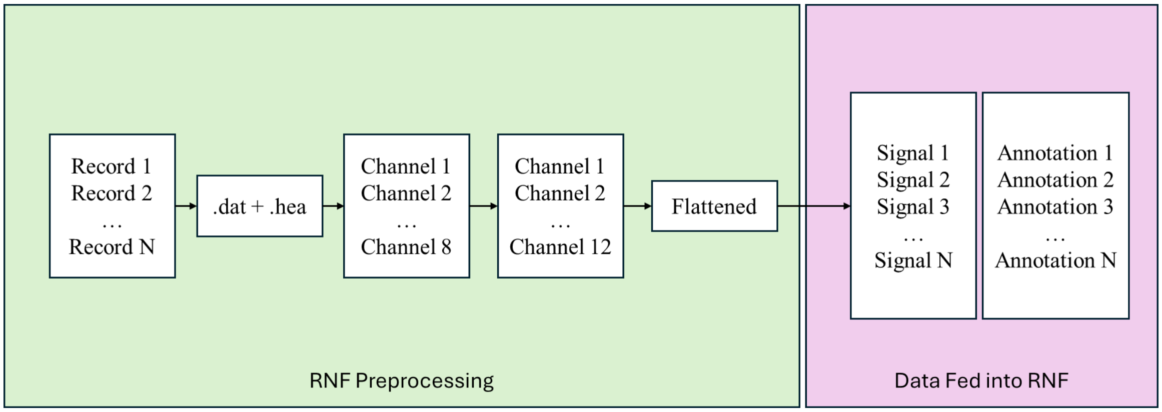


Figure 3.1.1: RNF Preprocessing

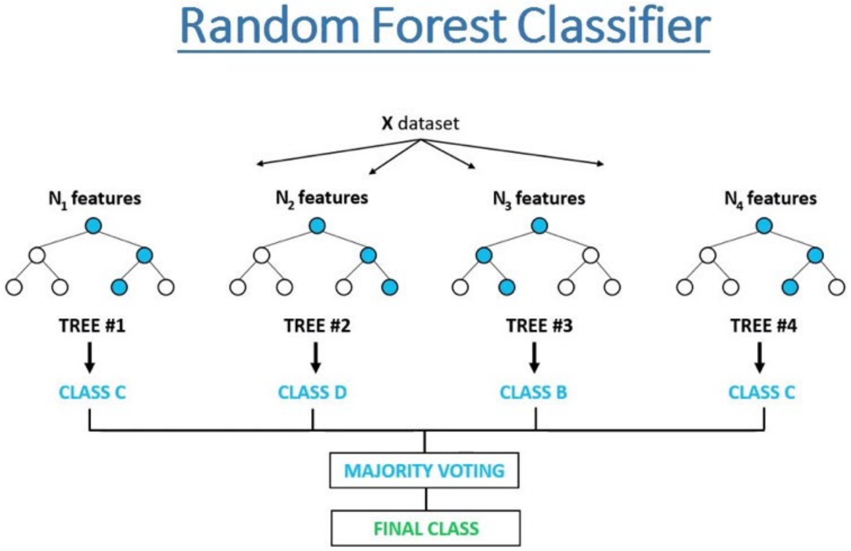


Figure 3.1.2: Graphic of Random Forest classifier system

3.2: Convolutional Neural Networks

In Senior Design I, our advisor, Dr. Joseph Picone, directed our attention to a published paper titled, *"Automatic diagnosis of the 12-lead ECG using a deep neural network,"* authored by Antonio H. Ribeiro et al. This study presents a comprehensive exploration of a complex convolutional neural network (CNN) for the detection of cardiovascular diseases using electrocardiogram (ECG) data. To train their CNN, the authors utilized a large dataset comprised of 2,322,513 million labelled exams sourced from the Telehealth Network of Minas Gerais

(TNMG). (Ribeiro, 2020). This dataset is gathered from a diverse pool of 1,676,384 different patients of 811 counties within the state of Brazil. (Ribeiro, 2020). Given the diversity dataset, our team reached out to the authors to request permission for utilizing the same data in our deep neural network, and we were able to receive their consent.

After conducting research about neural networks, our team opted to pursue a convolutional neural network as our second system to evaluate. Convolutional Neural Networks (CNN) are a class of deep learning algorithms that are specifically designed for processing images and time-series data. The architecture of CNNs is characterized by convolutional layers, which extract features from input data through the application of filters or kernels. These layers are typically followed by pooling layers, which ultimately down sample the feature maps to reduce computational complexity and extract the most important features. The main benefit of CNNs is their ability to automatically learn hierarchical representations of features directly from the raw data that it is given. The overall structure of CNNs make them optimal for wide ranges of applications, but most importantly, medical diagnoses.

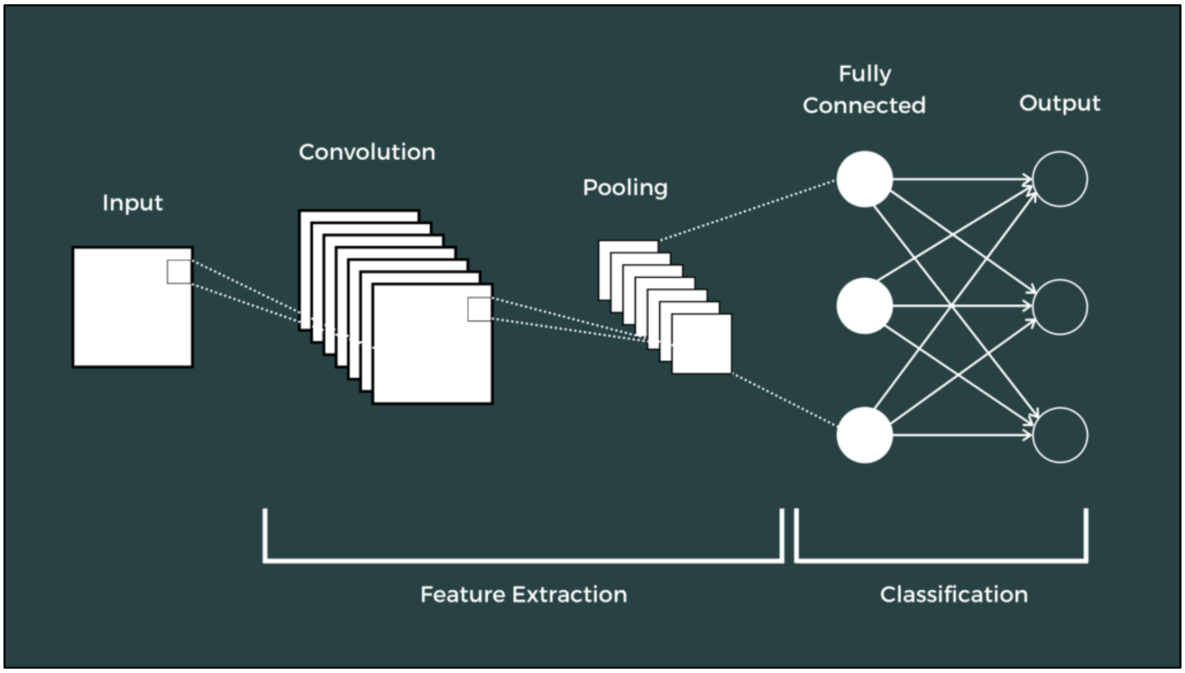


Figure 3.2.1: Graphical representation of CNN structure

Section 4: Engineering Design

4.1: Fully Balanced Dataset

When analyzing the full dataset provided by the published paper, the team found that the full dataset is biased heavily towards the diagnosis of *No Disease*. Out of the 2.3 million exams, *No Disease* takes up approximately 88.50% of the full data set, while the other six cardiac abnormalities take up the remaining 11.50%. In other words, one class is being overemphasized in the training process, which ultimately decreases the fairness in cardiovascular disease detection amongst the remaining classes. A model trained on this full dataset would lead to a heavily unbalanced model that may detect *No Diseases* more frequently than the other cardiac abnormalities. By cutting down on the 2.3 million exams to create a fully balanced dataset, the model will be trained on equal exam amounts of each disease, which would lead to less bias towards one singular classification. Furthermore, utilizing a fully balanced dataset would reduce the possibility of undertraining and overtraining one specific disease.

Therefore, to create the fully balanced dataset, we found the total amount of instances of each disease within the 2.3 million data set. The following findings were:

Disease	Disease Prevalence
1dAVb	35,759 (1.5%)
RBBB	63,528 (2.7%)
LBBB	39,842 (1.7%)
SB	37,949 (1.6%)
AF	41,862 (1.8%)
ST	49,872 (2.1%)
No Disease	2,053,701 (88.6%)

Table 4.1.1: Cardiovascular disease prevalence of full dataset

Analyzing *Table 1,* 1st Degree AV Block (1dAVb) has the lowest prevalence of 35,759 instances in the full dataset. Therefore, in the fully balanced dataset, the remaining five cardiovascular diseases (RBBB, LBBB, SB, AF, ST, No Disease) would be truncated down 35,759 instances, so that each class would have the same amount of training exams and can now be trained in a non-biased manner.

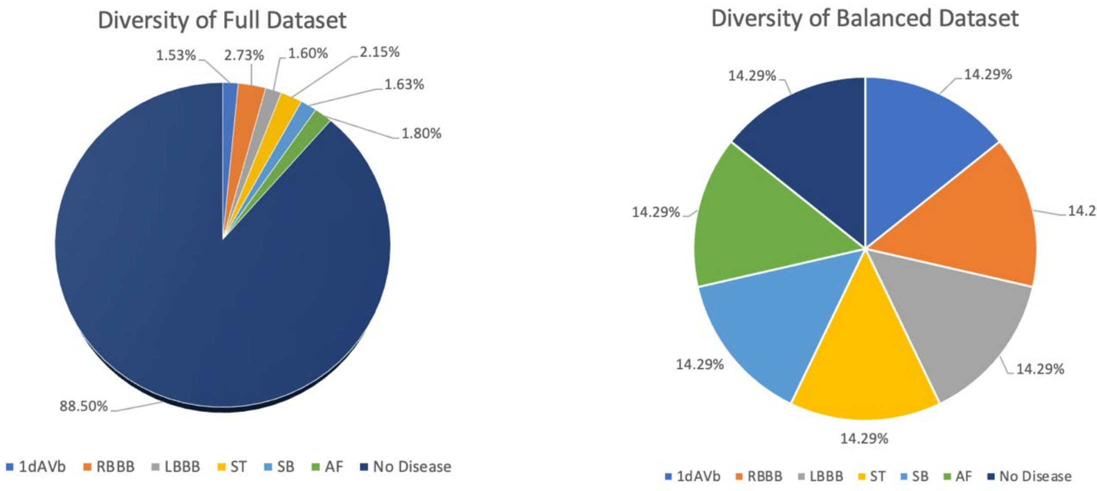


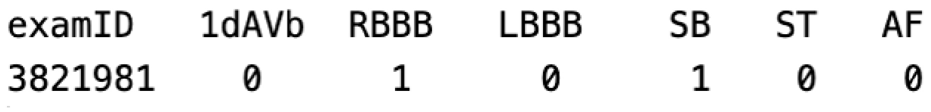
Figure 4.1.1: Comparison of unbalanced versus balanced dataset

While training with a balanced data set promotes fairness, it also results in a decrease in the number of training examples compared to an unbalanced data set. This decrease did raise concerns within the group regarding model undertraining, considering that the general principle in machine learning is that higher volumes of input data lead to better model performance. Despite this concern, we anticipate minimal impact on F1 scores, given the already low prevalence of diseases compared to non-disease instances in the full data set. After careful consideration, our team has chosen to proceed with training the model using a balanced dataset approach.

4.2: Confusion Matrices

4.2.1: Binary Annotations

Once the chosen machine learning model is trained, we need to test the performance of the classifier. The first step in assessing the performance of our cardiovascular disease detection machine learning models is to run a set of predictions over a validation dataset. A validation dataset is a set of ECG tracings that also include the ground truth binary disease annotations associated with each exam. An example of a binary disease annotation can be seen below in Figure 4.1:



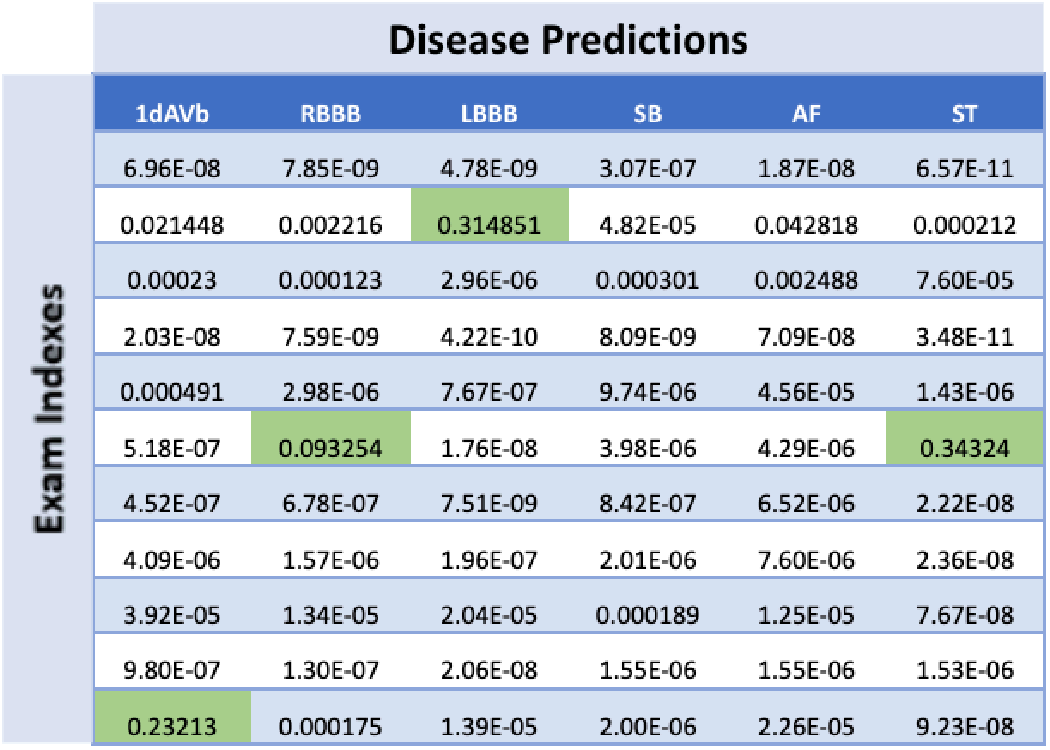
```
examID   1dAVb  RBBB   LBBB    SB   ST   AF
3821981    0      1      0      1    0    0
```

Figure 4.2.1: Example Binary Disease Annotation String

Each column of the binary string represents the state of one of the six cardiovascular diseases our model is going to be trained to detect. In Figure 2.1, there is a one in the column labeled RBBB (short for Right Bundle Branch Block) and the column labeled SB (short for Sinus Bradycardia), with zeros in every other column. This means the diseases present in this ECG exam are Right Bundle Branch Block and Sinus Bradycardia.

4.2.2: Example of Confusion Matrices

Using the binary disease annotations, and their corresponding ECG tracings, we can run a machine learning prediction across a validation dataset. In these predictions, the machine learning model will generate a matrix of float values representing the likelihood that the disease is present in each exam. After this, the program performs what is called “thresholding”, where the program selects the cutoffs for these float values where the prediction will either be considered a positive prediction (1) or a negative prediction (0). An example of one of these float matrices and its threshold values can be seen below:

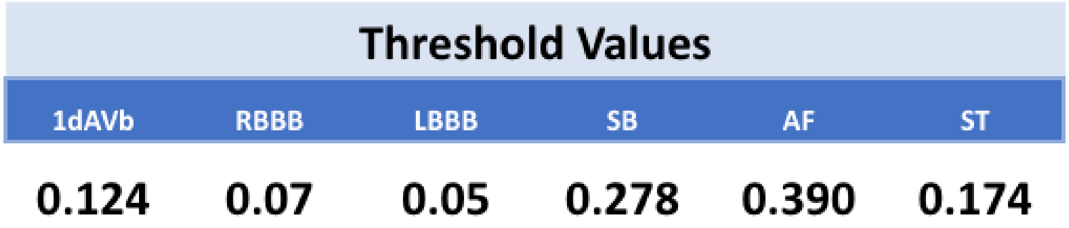


Disease Predictions

Exam Indexes

1dAVb	RBBB	LBBB	SB	AF	ST
6.96E-08	7.85E-09	4.78E-09	3.07E-07	1.87E-08	6.57E-11
0.021448	0.002216	0.314851	4.82E-05	0.042818	0.000212
0.00023	0.000123	2.96E-06	0.000301	0.002488	7.60E-05
2.03E-08	7.59E-09	4.22E-10	8.09E-09	7.09E-08	3.48E-11
0.000491	2.98E-06	7.67E-07	9.74E-06	4.56E-05	1.43E-06
5.18E-07	0.093254	1.76E-08	3.98E-06	4.29E-06	0.34324
4.52E-07	6.78E-07	7.51E-09	8.42E-07	6.52E-06	2.22E-08
4.09E-06	1.57E-06	1.96E-07	2.01E-06	7.60E-06	2.36E-08
3.92E-05	1.34E-05	2.04E-05	0.000189	1.25E-05	7.67E-08
9.80E-07	1.30E-07	2.06E-08	1.55E-06	1.55E-06	1.53E-06
0.23213	0.000175	1.39E-05	2.00E-06	2.26E-05	9.23E-08

Figure 4.2.2: Example Prediction Float Matrix

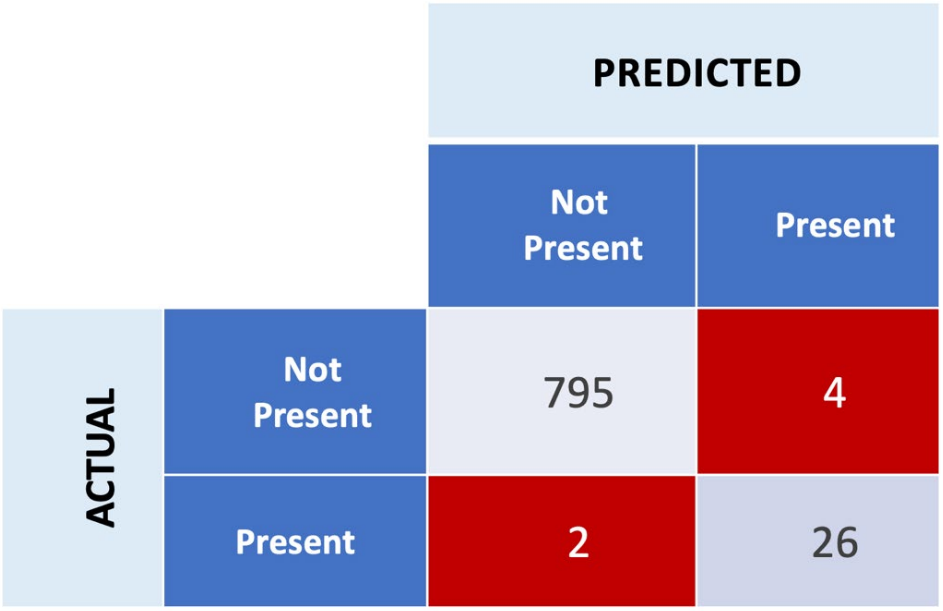


Threshold Values					
1dAVb	RBBB	LBBB	SB	AF	ST
0.124	0.07	0.05	0.278	0.390	0.174

Figure 4.2.3 Example Threshold Vector

In Figure 4.2, each row represents a specific ECG exam that the machine learning predicted over. Each column holds the prediction float value for each disease classification. The values highlighted in green are the values that exceed the threshold values (depicted in Figure 4.2.3) for their corresponding disease type and are labeled as positive predictions. All other values are labeled negative predictions. These threshold values are essentially generated by trial and error in a computer program, where the program will test different threshold values and select the ones with the best performance.

After the thresholding process, a program will mark whether the machine learning model was right or wrong for each exam. The output of this process is what is known as a confusion matrix. An example of a confusion matrix for one cardiovascular disease can be seen below in Figure 4.2.4:



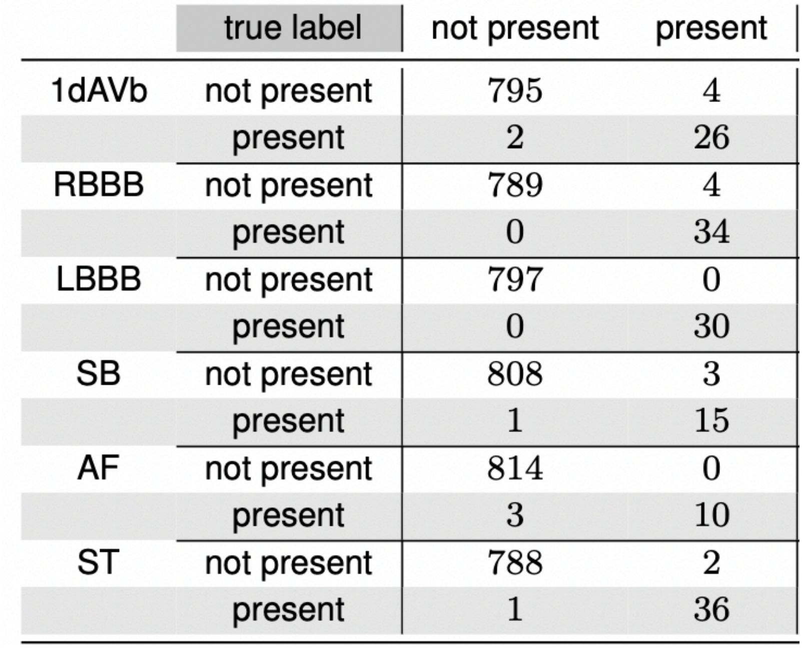
		PREDICTED	
		Not Present	Present
ACTUAL	Not Present	795	4
	Present	2	26

Figure 4.2.4: Sample Confusion Matrix for Single Disease

This confusion matrix depicts how the machine learning model predicted right and wrong across the validation dataset. This sample confusion matrix was generated from a validation dataset of 797 samples.

Each cell of the matrix represents the number of predictions that fall into each category. The first category, shown in the top left of the matrix, is the number of times the machine learning model correctly predicted that no disease was present in the validation dataset. This is known as a “True Negative” prediction, and there were 795 in this sample matrix. The next category, shown in the bottom left of the matrix, is the number of times the machine learning model incorrectly predicted that no disease was present, also known as a “False Negative” prediction. The next category, shown in the top right cell, is the number of times the machine learning model incorrectly predicted the disease was present in the exam. This is known as a “False Positive” prediction. Lastly, the category depicted in the bottom right cell is the number of times the machine learning model correctly predicted that the disease was present in the exam. This is known as a “True Positive” prediction.

The example confusion matrix shown in Figure 4.2.4 is for a single disease’s classifications. The full output of the machine learning predictions are confusion matrices for each disease combined into one matrix. An example of a full confusion matrix is shown in the following figure:



	true label	not present	present
1dAVb	not present	795	4
	present	2	26
RBBB	not present	789	4
	present	0	34
LBBB	not present	797	0
	present	0	30
SB	not present	808	3
	present	1	15
AF	not present	814	0
	present	3	10
ST	not present	788	2
	present	1	36

Figure 4.2.5: Full Confusion Matrix

In Figure 4.2.5, an example of a full confusion matrix is shown. This matrix contains the confusion matrices from each disease type, and they are stacked vertically.

4.3: Precision, Recall, and Specificity

4.3.1: Precision

Once the predictions are placed into a confusion matrix, the counts for each category of prediction are extracted to be used for the next set of calculations. Using set theory, three values are calculated from the confusion matrix. These three values are known as precision, recall, and specificity.

The first value to calculate from the confusion matrices is precision, which is “defined as the probability that an object is relevant given that it is returned by the system” (David E. Losada, 2005). In this case, a relevant object is a positive disease prediction, so precision represents the probability that the model will predict that the specified disease is present. Precision is calculated using the following equation:

$$p = \frac{TP}{TP + FP}$$

Equation 4.3.1: Precision Calculation Formula (David E. Losada, 2005)

In Equation 4.3.1, TP represents the number of True Positive predictions, while FP represents the number of False Positive predictions. Using this equation, precision can be interpreted as the “True Positive Rate”, or the number of True Positive predictions divided by the total number of positive predictions. This essentially tells us how accurate the machine learning model is for detecting positive disease categories.

4.3.2: Recall

The next value to calculate from the confusion matrices is recall, which describes the “probability that a relevant object is returned” (David E. Losada, 2005). Recall is calculated using the following equation:

$$r = \frac{TP}{TP + FN}$$

Equation 4.3.2: Recall Calculation Formula (David E. Losada, 2005)

In Equation 4.3.2, TP represents the number of True Positive predictions, and FN represents the number of False Negative predictions. This tells us how many positive disease cases the machine learning detected out of all positive disease cases of that disease type. This is helpful, as it essentially depicts the volume of the machine learning model’s positive disease predictions.

4.3.3: Specificity

The next value to calculate is specificity, which describes the accuracy of the machine learning model for identifying negative disease cases (Lekhtman, 2019). The formula for specificity can be seen below:

$$Specificity = \frac{TN}{TN + FP}$$

Equation 4.3.3: Specificity Calculation Formula (Lekhtman, 2019)

In Equation 4.3.3, TN represents the number of True Negative predictions for a single disease type, while FP represents the number of False Positive predictions. This essentially divides the number of True Negative predictions by the total number of negative classifications for the disease. This is similar to recall, but for negative disease classifications, meaning specificity depicts the volume of the machine learning model's negative disease predictions. Specificity is very helpful, as recall and precision deal with the model's performance when it comes to positive disease predictions, while specificity tells us the model's performance for its negative disease predictions.

4.4: F1 Scores

4.4.1: Introduction to F1 Scores

While these values are useful on their own, precision and recall can be used to calculate what is known as the F1 Score of the machine learning model. A F1 Score is a number ranging from 0 to 1 that describes the performance of the machine learning model. A zero is the worst-case score, meaning the model did not get a single prediction correct, while a one is a perfect score.

4.4.2: Harmonic Means and F1 Score Calculations

F1 Score is the harmonic means of precision and recall. A harmonic mean is type of average that is preferable when dealing with values that represent rates (Harmonic Mean, 2024). The formula for a harmonic mean can be seen below:

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \cdots + \frac{1}{x_n}}$$

Equation 4.4.1: Harmonic Mean Formula (Harmonic Mean, 2024)

In Equation 4.4.1, H represents the harmonic mean of a dataset $(x_1, x_2 x_3, \dots, x_n)$ containing n points. This formula is the number of points in the dataset divided by the sum of the reciprocals of the datapoints. Taking the harmonic mean of precision and recall yields this equation for F1 Score:

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

Equation 4.4.2: Harmonic Mean of Precision and Recall (Frank, 2023)

This can be rationalized to yield this formula:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Equation 4.4.2: F1 Score Formula (Frank, 2023)

Using the definitions of precision and recall, we can better understand what an F1 Score means. Precision depicts how often the machine learning model correctly detects positive disease cases, and recall depicts how often model detects disease cases out of all the positive disease cases. Precision can be interpreted as the “quality” of the model’s disease predictions, while recall can be interpreted as the “quantity” of the model’s disease predictions. Taking the harmonic mean of these values essentially generates a metric for evaluating both the volume of the model’s predictions and the accuracy of the predictions. This is the main purpose of the F1 Score: to represent the machine learning model’s performance for identifying one of the six disease types as a single number that is both intuitive and accurate.

When assessing the performance of a machine learning model across a validation dataset, an F1 Score will be generated for each of the six disease classifications. This will tell us how well the model detects each disease and will help us to identify any issues in our training data, such as overtraining.

Section 5: Evaluation - Test Methods & Results

To achieve alignment with the outcomes of the referenced study, extensive testing is imperative in this project's pursuit. Utilizing a deep neural network underscores the necessity for thorough examination due to its inherent complexity and non-linear nature. Variability in results, arising from factors like network architecture, hyperparameters, and dataset characteristics, mandates comprehensive evaluation. Rigorous testing protocols, encompassing closed-loop testing and evaluation dataset analysis, are crucial to validate the model's generalizability and performance consistency. Techniques such as dropout and regularization serve to counter overfitting and enhance model stability. Through meticulous testing and validation procedures, this project endeavors to attain results closely mirroring the referenced study, thereby bolstering confidence in the reliability of the developed model.

5.1: Closed Loop Testing

To test our machine learning models, there are two routes that can be taken to analyze the trained model's performance. One way is via closed loop testing, which essentially consists of testing the model's performance on the same data that it was trained on. By doing so, we can analyze the level of "absorption" that the model picked up on from training on its input data. On another note, closed loop testing allows us to pinpoint which portion of the training process needs attention. For example, by analyzing this table below, the utilization of closed loop testing can be demonstrated:

Disease Type	5k	10k	15k	20k
1dAVB	0.320	0.900	0.910	0.910
RBBB	0.910	0.970	0.970	0.970
LBBB	0.920	0.970	0.970	0.970
SB	0.640	0.960	0.960	0.960
AF	0.750	0.970	0.960	0.960
ST	0.380	0.930	0.930	0.940

Table 5.1.1: Closed Loop Testing as Function of Exams

In this table, we can see that for 10k, 15k, and 20k exams, the average F1 scores for these categories are very high considering that the value of 1.00 is the highest possible for a F1 score. Yet, for 5k, the F1 scores are unstable and low. By testing these models on their own training data, we can conclude that 10k, 15k, and 20k trained models have strong capabilities to diagnose the six diseases: 1dAVB, RBBB, LBBB, SB, AF, and ST due to their high absorption of the data. In terms of 5k, these values are likely a result of a very small sample size – therefore, the models for 5k were not able to absorb enough information.

In conclusion, by utilizing closed loop testing on our project, we can analyze our trained models in an alternative perspective that allows us to observe how much data a trained model is able to pick up on.

5.2: Evaluation Data Set Testing

When testing a trained model on an evaluation dataset, we are essentially testing the model on data that the model has never seen before. By doing so, the true cardiac disease diagnosing ability of the trained model can be computed and analyzed. In other words, with closed loop testing, the model is being tested with information it has been exposed to before. With evaluation testing, the model is being tested with information it has never been exposed to before, allowing for the model to be evaluated in terms of its diagnosing accuracy. For example, by analyzing this table below, the utilization of evaluation dataset testing can be demonstrated:

Disease Type	5k	10k	15k	20k
1dAVB	0.090	0.580	0.570	0.560
RBBB	0.840	0.830	0.830	0.820
LBBB	0.880	0.900	0.870	0.910
SB	0.400	0.760	0.770	0.710
AF	0.040	0.040	0.050	0.030
ST	0.320	0.100	0.090	0.090

Table 5.2.1: Evaluation Dataset Testing as a Function of Exams

In this table, we can see that the F1 scores for all exam sample sizes are not as close to 1.00 as closed loop testing was. The large fluctuations in F1 scores can be due to several reasons, including perhaps the respective sample size is still too low. But the main factor to why these numbers are not as high as closed loop testing is because the model is being introduced to completely new ECG exams and must make blind predictions. Evaluation datasets often contain a wider range of examples or ECG instances that the model hasn't encountered during training. This blind variability can challenge the model's ability to generalize its learned patterns to unseen/new ECG data, which can lead to lower performance. In conclusion, by utilizing evaluation dataset testing on a trained model, we can analyze the model's ability to predict the six cardiac diseases when presented with new data.

5.3: Preliminary Random Forest Testing Results (Evaluation Set)

To test the implementation of the Random Forest classifier, we utilized an approach identical to the testing methods used in Section 5.2. Four Random Forest classifiers were trained using 5k, 10k, 15k, and 20k ECG exams, respectively. These four classifiers were then tested using the evaluation dataset, yielding the following results:

Disease Type	5k	10k	15k	20k
1dAVB	0.047	0.060	0.069	0.050
RBBB	0.052	0.036	0.053	0.056
LBBB	0.097	0.055	0.122	0.078
SB	0.040	0.042	0.056	0.044
AF	0.035	0.038	0.041	0.030
ST	0.043	0.045	0.015	0.000

Table 5.3.1: Random Forest Testing Results (F1 Scores)

The F1 scores shown in the table are exceptionally low, and there is very little improvement in performance as the number of exams in the training dataset increases. Based on these results, we deemed our Random Forest classification system not ideal for this project and decided to test the Convolutional Neural Network.

5.4: Preliminary CNN Testing Results (Evaluation Set)

To test the performance of the Convolutional Neural Network (CNN), we trained four CNNs using the strategy outlined in Section 5.2. Then, we tested each CNN over the evaluation set and calculated the F1 score for each disease, for each CNN, yielding the following results:

Disease Type	5k	10k	15k	20k
1dAVB	0.090	0.840	0.880	0.400
RBBB	0.580	0.830	0.900	0.760
LBBB	0.570	0.830	0.870	0.770
SB	0.560	0.820	0.910	0.710
AF	0.090	0.840	0.880	0.400
ST	0.580	0.830	0.900	0.760

Table 5.4.1: Convolutional Neural Network Testing Results (F1 Scores)

These results are much more encouraging compared to the Random Forest results shown in Table 5.3. Although there was a drop in performance as the number of exams in the training set increased from 15k to 20k, there is still a general trend of improvement as the number of exams increases. Based on these results, we decided to proceed using CNN as our chosen classifier, and train a model using the full 250k exam, balanced dataset.

5.5: Final CNN Results

After training a CNN using the full 250k exam dataset containing balanced instances of each disease, we decoded the evaluation dataset, and calculated the F1 scores for each disease. The resulting F1 scores can be seen in the following table:

Model Architecture	**1dAVB**	**RBBB**	**LBBB**	**SB**	**AF**	**ST**
CNN	0.868	0.939	0.984	0.857	0.933	0.700

Table 5.5.1: Final CNN Evaluation Results (F1 Scores)

Comparing these results to our design criteria metrics yields the following table:

	Disease Type					
Classifier	1dAVB	RBBB	LBBB	SB	AF	ST
Cardio.	0.776	0.917	0.947	0.882	0.769	0.882
Emerg.	0.719	0.852	0.912	0.848	0.696	0.946
Stud.	0.732	0.928	0.915	0.750	0.706	0.873
DNN	0.897	0.944	1.000	0.882	0.870	0.960
Our CNN (250k)	0.868	0.939	0.984	0.857	0.933	0.700

Table 5.5.2: Final CNN F1 Scores vs. Design Criteria

The results from our CNN trained on 250k exams are strong for all disease classifications except for ST, which only has an F1 score of 0.700. This likely because in our dataset, occurrences of ST are often paired with instances of other disease as well, making it difficult to isolate this disease to sufficiently train the CNN. Our CNN's performance for all other disease types meets our non-negotiable design criteria of matching or exceeding the F1 scores of the cardiology residents, emergency residents, and medical students. Our CNN obtains a higher F1 score than the DNN published in "Automatic diagnosis of the 12-lead ECG using a deep neural network" when classifying AF. Other than for this disease, our CNN does not meet the performance of the study's published DNN.

Section 6: Standards & Specifications

6.1: Standards & Specifications

For our analysis, we will evaluate the performance of our machine learning model using F1 Scores, a widely accepted metric in assessing machine learning models. This approach aligns with the methodology of the study we are replicating, titled "Automatic diagnosis of the 12-lead ECG using a deep neural network" (2020), which has garnered significant recognition with over one hundred citations (National Library of Medicine, 2024). Moreover, recent machine learning studies like "Auto-detection of the coronavirus disease by using deep convolutional neural networks and X-ray photographs" (2024) and "Improving deep neural network generalization and robustness to background bias via layer-wise relevance propagation optimization" (2024) have also utilized F1 scores, highlighting its relevance and effectiveness in model evaluation. F1 scores offer a comprehensive evaluation by combining precision and recall into a single normalized metric, providing a succinct characterization of model performance.

Another standard procedure we intend to implement involves converting file types to adhere to the Institute of Image Signal Processing (ISIP) standard for medical signal storage. Initially provided in the form of .dat and .hea files, where .dat files contain binary data representing ECG voltage waveforms and .hea files hold crucial metadata like sampling frequency and signal duration, we plan to merge each pair into .edf files. These .edf files, standard for storing EEG signals at ISIP (N. Capp, 2018), offer the advantage of consolidating signal data and metadata into a single file format, thereby streamlining Python file input/output operations. Leveraging ISIP's NEURONIX server, equipped with specialized signal processing tools like the NEDC EEG Annotation System and the EDF Browser (ISIP, Open Source EEG Resources, 2024), tailored for .edf files, enhances our ability to manipulate and analyze ECG signals efficiently.

Moreover, to ensure consistency and clarity in our project, all code will adhere to ISIP laboratory standards. This entails meticulous commenting and adherence to guidelines for function, file, and variable naming (ISIP, Programming Style, n.d.). Additionally, we will provide detailed documentation outlining the purpose and functionality of all scripts, facilitating portability and comprehension for future users.

Section 7: Project Costs

To train the array of machine learning models utilized in our research project, we leveraged the NEURONIX cluster provided by the Institute of Image and Signal Processing at Temple University. This high-performance cluster is housed within the Joint Data Center (JDC) at Temple University's TECH Center. While precise energy consumption figures are unavailable due to limited access to the university's financial data, we can approximate the associated costs by considering the GPUs employed in the process. Each node in the cluster contains four GPUs of the same model:

Node	GPU Model
nedc_007	NVIDIA GeForce GTX 1070
nedc_008	Tesla P40
nedc_011	NVIDIA GeForce RTX 2080
nedc_012	NVIDIA A40

Table 7.1: Node GPU's

Most of the training was done on the node nedc_012. This node contained the fastest GPU's which were the NVIDIA A40.

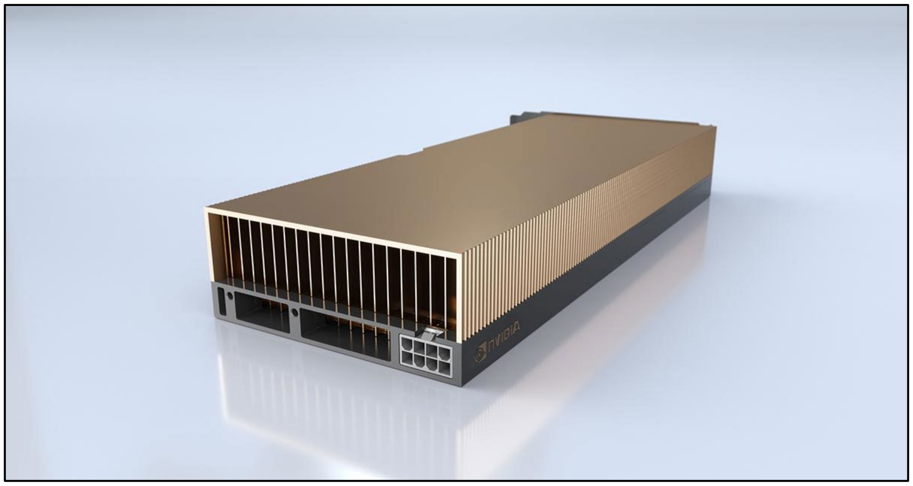


Figure 7.1: NVIDIA A40

The following are the GPU specifications in terms of their max power consumption gathered from the respective manufacturer's websites:

GPU Model	Max Power Consumption (kWh)
NVIDIA GeForce GTX 1070	0.150
Tesla P40	0.250
NVIDIA GeForce RTX 2080	0.215
NVIDIA A40	0.300

Table 7.2: GPU Power Consumption

According to the U.S. Bureau of Labor Statistics, the average price per kWh in Philadelphia for February 2024 was $ 0.201. With the architecture used to generate our 250k exam balanced model, it took approximately 25 hrs. (Statistics, 2024)

$$Cost\ (\$) = (Price\ per\ kWh) * (4\ GPUs\ per\ Node) * (GPU\ Power\ Consumption) * (Program\ Run\ Time)$$

Equation 7.1: Cost Estimate Equation

GPU Model	Total Cost ($)
NVIDIA GeForce GTX 1070	2.90
Tesla P40	4.83
NVIDIA GeForce RTX 2080	4.15
NVIDIA A40	5.79

Table 7.3: Cost of 250k Exam Model

The overall expense incurred in training these machine learning models, factoring in GPU power consumption, remains relatively modest. However, it's important to acknowledge that these figures do not encompass the power consumption associated with operating the compute nodes and clustering infrastructure itself. Furthermore, it's worth highlighting that the models underwent multiple training iterations, with a significant portion of server time allocated to software development tasks. Actual model training constituted only a fraction of the total server runtime.

Section 8: Summary and Conclusions

This senior design project aimed to investigate various approaches and optimize existing solutions for detecting cardiovascular diseases in 12-Lead ECG signals. Our implementation of the studies pre-existing Convolutional Neural Network (CNN) demonstrated superior performance compared to 4th year cardiologist residents, 3rd year emergency residents, and medical students, as measured by F1 scores. Despite using a balanced dataset containing 250,000 exams, significantly smaller than the 2 million exams used in previous studies, our CNN still surpassed the performance of medical professionals. While we did not outperform the studies' Deep Neural Network (DNN), further experimentation as well as training different permutations could potentially yield even better results. The inherent flexibility of DNNs means that they can reach convergence through various pathways, suggesting that retraining the model from scratch might lead to higher F1 scores than those reported in this paper. Additionally, our analysis revealed a significant underrepresentation of sinus tachycardia exams in the dataset, likely due to its frequent co-occurrence with other diseases. Addressing this imbalance by training the model with more sinus tachycardia data could further improve the accuracy of our results.

Looking ahead, we envision the integration of machine learning programs, like the one developed in this project, into low-cost ECG machines. We would also like to see, provided the data exists, more cardiovascular diseases to be added to the prediction aspect of this software. This software currently only covers a small subsect of the current disease that exists in the heart. On top of this, we would like to see this software be furthered by incorporating other patient factors such as age, height, weight, and gender to better make accurate disease predictions. By doing so, we can advance toward our ultimate objective of delivering state-of-the-art healthcare to underserved populations in developing countries. This technological integration holds immense potential to democratize access to accurate and efficient cardiovascular disease diagnosis, even in resource-constrained settings where traditional medical infrastructure may be lacking. As we continue to refine and optimize these machine learning algorithms, their deployment in affordable ECG devices could significantly improve healthcare outcomes and contribute to reducing global health disparities.

Works Cited

David E. Losada, J. M.-L. (2005). *Advances in Information Retrieval.* Santiago de Compostela, Spain: 27th European Conference on IR Research.

Frank, E. (2023, April 29). *Understanding the F1 Score*. Retrieved from Medium: https://ellielfrank.medium.com/understanding-the-f1-score-55371416fbe1

Harmonic Mean. (2024, January 24). Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/harmonic-mean/

https://sdgs.un.org/goals. (2024). Retrieved from United Nations, Department of Economics and Social Affairs.

Lekhtman, A. (2019, August 5). *Data Science in Medicine-Precision and Recall or Specificity and Sensitivity?* Retrieved from Medium: https://towardsdatascience.com/should-i-look-at-precision-recall-or-specificity-sensitivity-3946158aace1

NVIDIA. (2024). *GeForce*. Retrieved from GeForce GTX 1070 Specifications: https://www.nvidia.com/en-gb/geforce/graphics-cards/geforce-gtx-1070/specifications/

NVIDIA. (2024, April). *GeForce RTX 20 Series*. Retrieved from NVIDIA: https://www.nvidia.com/en-us/geforce/20-series/

Ribeiro, A. R. (2020). Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nature Communications*.

Statistics, U. B. (2024, April 29). *Average energy prices for the United States, regions, census divisions, and selected metropolitan areas*. Retrieved from Bureau of Labor Statistics: https://www.bls.gov/regions/midwest/data/averageenergyprices_selectedareas_table.htm

TECHPOWERUP. (2024). *NVIDIA Tesla P40*. Retrieved from TECHPOWERUP: https://www.techpowerup.com/gpu-specs/tesla-p40.c2878