**Abstract (281 words 🡪 too much 🡪 limit to 250 words).**

The high-level objective of our project is to generate synthetic ECG signals on embedded systems using generative adversarial networks. Indeed, as cardiovascular diseases are the leading cause of death in the world combined to the growing use of machine learning for diagnostics there has been an increase in the need of ECG datasets which can be difficult to acquire due to privacy and accessibility reasons. To achieve this goal, we implemented a generative machine learning model to produce ECG signals. Then, using this model, we converted it to a microcontroller adapted model using tiny machine learning (TensorFlow Lite). The next step is then to upload the model onto an Arduino BLE and compare the results to the literature. Prior work has been done on GAN-generated ECGs and has produced encouraging results but there has yet to be an implementation on microcontrollers. The generator model that was trained has produced similar result to ECG signals with 0.06364 mmd considering the size of our dataset and limited size capacity for our chip. Our initial generator mode size was 1891.773 KBs and after converting it to a TensorFlow Lite model, the size was reduced to 89.18 KBs fitting our maximum size of 1MB for our microcontroller. We believe there are still a lot of improvements to be done for this project that could lead to even more promising results. The limited size of our dataset can be expanded for a better model training as we only have 565 signals that were used. The next steps would then be to train more models for specific types of ECGs and continue working on memory management as the capacity of our microcontroller is limited.

1. **Introduction (878 words)**

Cardiovascular diseases have been and still are the leading cause of death in the world. In 2015, it was reported that approximately 32.1% of the deaths in the world were due to cardiovascular diseases [4]. These numbers are likely to continue increasing, indeed more and more cases have been reported among younger people. Factors such as increased stress and psychological issues can explain such a tendency. For cardiovascular diseases, electrocardiograms are the golden standard in clinic for diagnostics. Through electrodes placed on the thorax, electro-physiological signals can be obtained and can help doctors diagnose heart diseases and predict heart failures. There can be a lot of data generated from these ECGs as there can be up to 12 leads and recordings can go for extended periods of time and analysis of this data can be very time consuming.

Machine learning is therefore employed to facilitate this analysis. Indeed, deep learning and artificial has seen significant advances in the healthcare industry. They allow to lower the costs as well as increase the efficiency of medical diagnostics. Indeed, simple PQRST annotation can be extremely time consuming simply on a 12-lead ECG recording of 10 minutes. Knowing that the average heart rate is sixty beats per minute that can be up to 7,200 PQRST annotations to do, which would take forever for a human. In 2019, the predicted market for artificial intelligence in 2023 was 97.9 billion dollars. However, for this to happen, we rely on datasets from electronic health records. The results have proved to be very efficient and not only for cardiovascular diseases. Using advanced algorithms, researchers have managed to reduce misdiagnosis of cancer by 85%. For breast cancer, the CNN models developed by Google have a 99% accuracy diagnostic compared to a low 38% accuracy by doctors. Machine learning on retinal images has also allowed to diagnose diabetic retinopathy and there are many more examples [5]. This proves that machine learning and deep learning techniques have and will have an increased importance in the coming years in medical diagnostics.

As we saw, the use of AI has increased over the past years and will continue doing so allowing clinicians to diagnose diseases more accurately. For this, a large amount of data is necessary. However, medical data is of highly sensitive nature. The access and publishing of this data is highly regulated as to avoid any potential leaks. In the past years, there has been proof of very important medical data leaks. KoczKodaj et. Al [1] have shown that there is a steady increase in security breaches of data throughout processing systems and that these threats will keep rising. In 2017, a single leakage of data lost 1.3 billion records! So this data must be anonymized especially with the updates in General Data Protection Regulation in the EU that have tightened the rules regarding data protection. A common method to protect the data is de-identification. However, it has been shown that de-identification does not guarantee the protection of privacy as features can be extracted and linked to a person. [2]. Indeed, by simply knowing the zip code, the birthday and the sex of a person, it is enough to identify 87% of the citizens [3]. Therefore, the lack of datasets complicates the training of machine learning models and research is slowed down.

The combination of too few datasets and data protection issues, has oriented research towards alternatives and more particularly synthetically generated ECGs. There has been a lot of work done in that field with encouraging results. The objective of this project is not only to train an efficient model but also to make a sufficiently small model that can fit on a microcontroller (Arduino BLE). Indeed, an embedded solution for ECG generation would bypass the existing expensive techniques that are currently used in practice. For this, we have first trained a generative machine learning model to produce ECG signals. We then converted the model using tiny machine learning making it small enough to fit onto the microcontroller. Finally, we will compare the results to the current literature.

1. **Relevant Prior Work**

As stated previously, a lot of work has been done in ECG generation and more specifically using GANS. Before going through the state of the art and literature results of these techniques, it is important to introduce the theory between generative adversarial networks.

Like the name suggests, generative adversarial networks implement two competing neural networks: a generator and a discriminator. The generator is responsible for generating the synthetic data whereas the discriminator is in charge for classifying real data from fake data generated by the generator. As seen on the figure below, the generator uses random noise as an input to begin with. With this random noise, the generator will generate data (a vector of values in our case) and feed it into the discriminator. The discriminator’s role is to correctly label this synthetic data as fake. However, the discriminator is also fed real electrocardiogram signals from a dataset that is used for training and the discriminator will need to label this data as real. At the beginning, the discriminator will easily classify the synthetic data and will feed back the information to the generator so that the generator can update its parameters and improve its model. This loop will continue recursively. The improvement of the generator’s model is based on the value function below. D(x) represents the probability of the discriminator to correctly classify real ecg signals coming from the dataset. G(z) represents the synthetic ECG signals generated by the generator. D(G(z)) is the probability of classifying the synthetic generated data as real. The recursiveness of the training is based on this value function. There is a competing optimization for both models until we converge to a point where the discriminator cannot tell the fake data from the real data anymore. Mathematically, the discriminator will want to maximize the probability of correctly classifying real data as real (max D(x)) and to maximize the probability of correctly classifying the synthetic data as fake (max 1-D(G(z))). The generator model, however, tries to minimize the probability of the discriminator to correctly classify the synthetic data as fake, it tries to fool the discriminator into thinking that the fake data comes from the dataset (min 1-D(G(z))).

Diagram

Description automatically generated

*GAN model [4]*



*Value function for GAN mode [4l*

This means that we will need a neural network for both the generator and discriminator. In the literature, there are different ways of implementing these models and we will go through several examples and the results of their ECG generation.

Debapriya et. Al [5] developed a GAN based method for biomedical signal generation including ECG signals. Their work was based on a Bidirectional Low Short Term Memory (BiLSTM) neural network-based generator. For their discriminator they used a more traditional convolutional neural network. Fei Zhu et. Al [4] combined a BiLSTM generator model with a CNN discriminator model for ECG generation only whereas Vajira Thrambawifa et. Al [3] used two CNN models for both the generator and discriminator model.

For the evaluation of their models, the following metrics were computed between the synthetic signals and original signals:

* Mean Absolute Error (MAE)
* Root Mean Square Error (RMSE)
* Percent Root Mean Square (PRD)

All the previous results are compared in the table below.

|  | MAE | RMSE | PRD |
| --- | --- | --- | --- |
| Debapriya et. Al [5] | 0.218 | 0.126 | 6.343 |
| Fei Zhu et. Al [4] | No reported | 0.276 | 66.408 |
| Thrambawita et. Al | Not reported | Not reported | Not reported |

*Metric Results*

These results will be compared to the model generated for our project in the results section.

Although ECG generation has been studied previously there has yet to be an implementation on microcontrollers and therefore the generator model will be subject to some size constraints and the results will therefore be influenced by this constraint.

1. **Technical Approach**

In this section we will discuss our technical approach to the project’s objectives.

* 1. **Dataset**

Training the models requires a set of real ECG signals. For our project we acquired our dataset from the biomedical signal bank PhysioNet and more particularly the “MIT-BIH Arrhythmia Database”. The database is made of 47 subjects with 48 half-hour recordings (one subject was recorded twice) digitized at 360 samples per second.

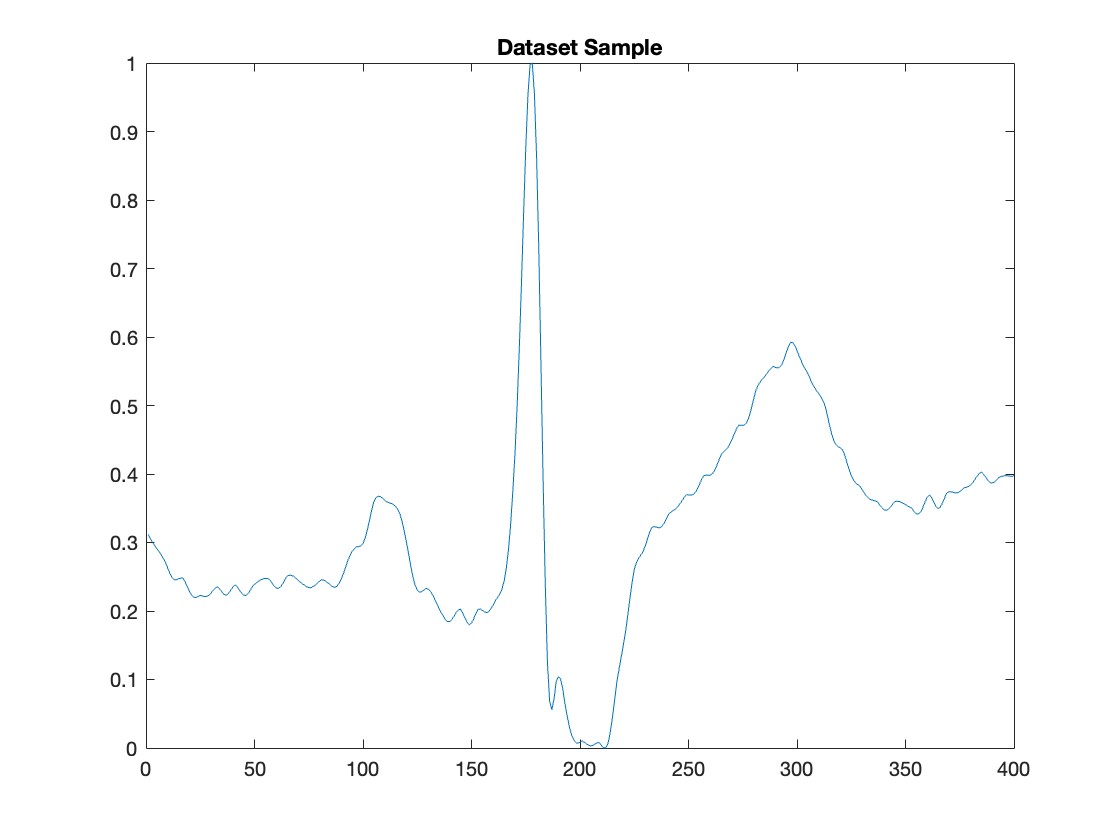
Our models were initially based on the GAN developed in the article by Fei Zhu et. Al [4]. This meant that some preprocessing needed to be done on the dataset. For each subject, a 10 second recording was extracted. The signals on the database had been amplified with a 200 gain so the signals were normalized to have raw signals. To match the literature processing, the signals were downsampled to 125Hz yielding 1200 sample signals. From these 48 signals, 100 sample signals were extracted. The final dataset used for training was made of 565 sample recordings (a lot of data had to be removed as it was not exploitable).

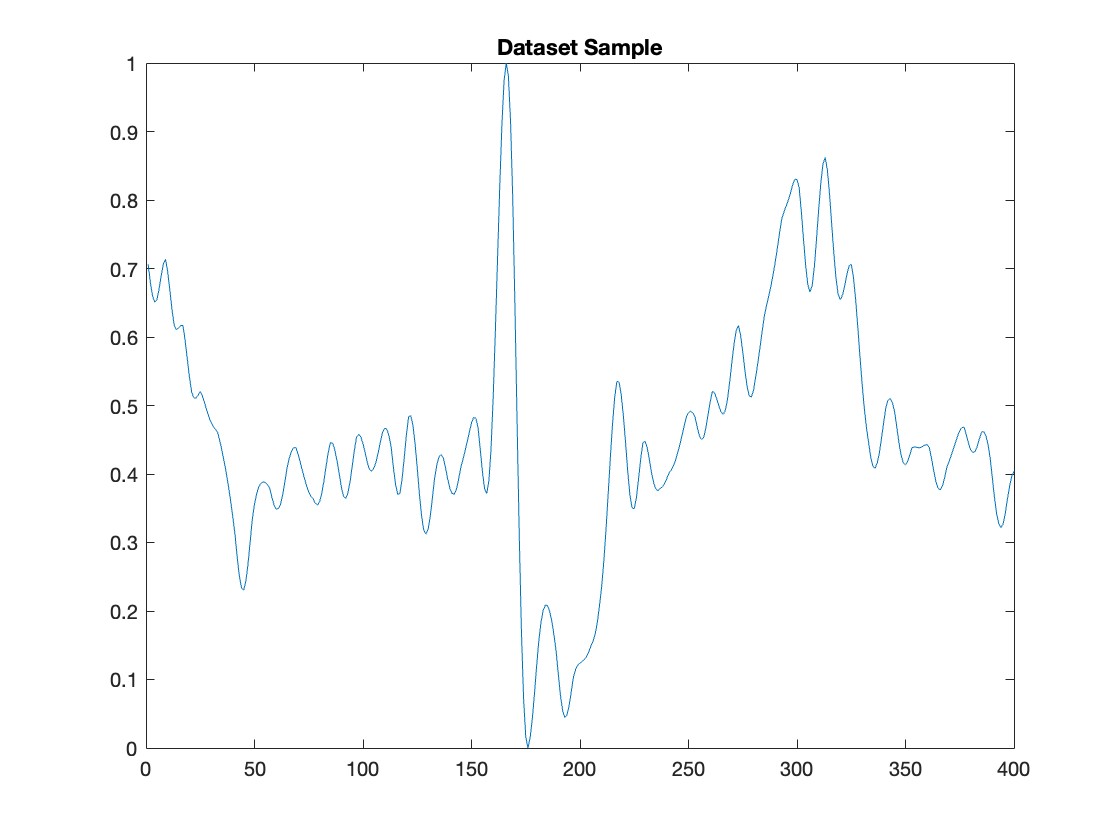
* 1. **Models**

Our models were based on the work of Fei Zhu et. Al [4] and adapted to our objectives. Their generator model was a bidirectional low short term memory combining the BiLSTM and four convolutional neural networks. Our objective required to convert the model to a microcontroller sized model and the functions that we use are not applicable for bidirectional networks so the first step was to remove this layer for the generator model. Filter sizes were then reduced by half to reduce model size. The total number of parameters for our generator model was 44,281.

As our discriminator model was only used for training, we were not bounded by size. We implemented the model similarly to the one used by our reference model but we adapted the input and output sizes to match our dataset.

**4. Results and evaluation**



****

Initial generator model size: 205.836 kB.

TensorFlow Lite model size: 89.19 KB.

**Metric Results:**

**MMD:** mean = 0.063644 min=0.000556 max=0.348375

**PRD:** mean=11.0991 min=5.0097 max=41.8445

**RMSE:** mean=0.3165 min=0.1515 max=0.6116

**5. Conclusion**

As seen previously, the model yields relatively satisfying results but when we compare our metrics to the literature, there is still room for significant improvements. While we initially struggled to convert the model to a satisfying size, we succeeded in doing so and ended up with a 89KB model. Although this was our initial objective, the next steps would be to improve our results by implementing the following steps.

The Arduino BLE 33 limits our model size to a maximum of 1MB, meaning that we still have a lot of room to improve our model and get better metrics while still managing to fit it on to the micro-controller.

Our dataset was not as extensive as the datasets that were used for training in literature. We limited our ECG dataset to a one 10-second window from each initial signal. We could exploit multiple seconds, minutes or even the whole 30 minutes of recording for each of the 48 signals and therefore greatly expand our dataset to a more efficient number for training. Training our model with significantly more samples will provide better results for our generator as the discriminator has much more examples to provide a better discrimination.

Our model was trained on an arrhythmia based database and therefore a lot of signals can be very unstable. During the training of our model, no separation between the pathologically-induced signals and the healthy ones was made. Our model did not account for the variability of the ECGs and therefore the discriminator will have a harder time defining a clear distinction between the synthetic signals and the dataset signals. More pre-processing can be done during signal extraction to focus the training on one specific type of heartbeat.

Overall, we succeeded in reaching our objective, being able to fit a GAN-based model onto a microcontroller able to generate ECG signals. However due to reasons discussed before, there are still a lot of improvements to be made and the future steps have already been identified.

**1.Massive Health Record Breaches Evidenced by the Office for Civil Rights Data**

**2.SYNTHESIS OF REALISTIC ECG USING GENERATIVE ADVERSARIAL NETWORKS**

**3.DeepFake electrocardiograms using generative adversarial networks are the beginning of the end for privacy issues in medicine**

**4..Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network**

**5..SynSigGAN: Generative Adversarial Networks for Synthetic Biomedical Signal Generation**

.