

IMPLEMENTACIÓN EN UN MICROCONTROLADOR DE LA INFERENCIA DE  
UNA RED NEURONAL CONVOLUCIONAL PARA LA DETECCIÓN DE  
ARRITMIAS CARDIACAS

JOSE LUIS CADENA TÉLLEZ  
JULIAN EDUARDO JEREZ VERA

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIA FISICO-MECANICAS.  
ESCUELA DE INGENIERIA ELECTRICA, ELECTRONICA Y  
TELECOMUNICACIONES  
BUCARAMANGA

2021

IMPLEMENTACIÓN EN UN MICROCONTROLADOR DE LA INFERENCIA DE  
UNA RED NEURONAL CONVOLUCIONAL PARA LA DETECCIÓN DE  
ARRITMIAS CARDIACAS

JOSE LUIS CADENA TÉLLEZ  
JULIAN EDUARDO JEREZ VERA

Trabajo de grado para optar al título de Ingeniero Electrónico

Director:

CARLOS ARTURO BOADA QUIJANO  
Magister en Ingeniería Electrónica

Codirector:

SAID DAVID PERTUZ ARROYO  
PhD en Ciencias de la computación

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIA FISICO-MECANICAS.  
ESCUELA DE INGENIERIA ELECTRICA, ELECTRONICA Y  
TELECOMUNICACIONES  
BUCARAMANGA

2021

## **AGRADECIMIENTOS**

Agradecer primero a Dios que siempre me guio en el camino de tomar las mejores decisiones para mi desarrollo personal. A mis padres, por el amor y el apoyo incondicional que siempre me han demostrado, que me permitieron culminar con este nuevo objetivo de mi vida personal. A mis hermanas, que a la distancia siempre estuvieron presentes con su apoyo en este proceso. Al resto de mi familia, compañeros y amigos que me deja la universidad, que de alguna u otra forma aportaron su granito de arena. A mi compañero de proyecto y al grupo de investigación en conectividad y procesamiento de señales (CPS), en especial a nuestro director y codirector que estuvieron presentes en este crecimiento intelectual. Finalmente, al profesor Carlos Fajardo que siguió de cerca nuestro trabajo, por el apoyo y todos los conocimientos brindados.

*Jose Luis Cadena Téllez*

Agradezco a Dios guiarme en este camino y proveerme de paciencia, dedicación y sabiduría para afrontar los retos y dificultades presentadas. A mi familia por apoyarme día a día, animándome a cumplir mis objetivos. A mis amigos y compañeros que me dio la universidad, por aportar de su tiempo, conocimiento, esmero y dedicación en el proceso que ha sido más ameno con ellos. A mi compañero de proyecto con quien formamos un equipo para llevar esto a cabo, a nuestro director y codirector que nos aportaron su apoyo y consejos. Al profesor Carlos Fajardo quien siguió de cerca nuestro proyecto y nos aportó de gran manera. A la Universidad Industrial de Santander, al grupo de investigación en conectividad y procesamiento de señales (CPS) por ser parte de mi crecimiento personal e intelectual. A todas las personas que de una u otra manera han aportado para alcanzar las metas propuestas, sin ustedes no habría sido posible.

*Julián Eduardo Jerez Vera*

## TABLA DE CONTENIDO

	pag.
INTRODUCCIÓN .....	9
1. DATA SET .....	12
2. RED NEURONAL CONVOLUCIONAL.....	13
3. IMPLEMENTACIÓN EN EL MICROCONTROLADOR.....	15
3.1. ADQUISICIÓN DE LA SEÑAL DE ENTRADA .....	15
3.2. FUNCIONES DE ARDUINO .....	17
4. RESULTADOS.....	19
5. CONCLUSIONES .....	22
BIBLIOGRAFIA.....	23

## LISTA DE TABLAS

	pag.
Tabla 1. Matriz de confusión del modelo.	19
Tabla 2. Medidas de rendimiento del modelo.	20
Tabla 3. Memoria utilizada en la implementación del modelo.	20
Tabla 4. Características de desempeño del microcontrolador.	21

## LISTA DE FIGURAS

	pag.
Figura 1. Modelo implementado. Modificación del Tiny Model.	14
Figura 2. Flujograma del modelo.	18
Figura 3. Procesos ejecutados en el microcontrolador.	18

## RESUMEN

**TÍTULO:** IMPLEMENTACIÓN EN UN MICROCONTROLADOR DE LA INFERENCIA DE UNA RED NEURONAL CONVOLUCIONAL PARA LA DETECCIÓN DE ARRITMIAS CARDIACAS\*

**AUTOR:** JOSE LUIS CADENA TÉLLEZ, JULIAN EDUARDO JEREZ VERA\*\*

**PALABRAS CLAVE:** ARRITMIAS CARDIACAS, DEEP LEARNING, INTELIGENCIA ARTIFICIAL, MICROCONTROLADOR, REDES NEURONALES CONVOLUCIONALES, SEÑALES ELECTROCARDIOGRÁFICAS.

**DESCRIPCIÓN:** Las enfermedades cardíacas tienen una de las más altas tasas de mortalidad en Colombia. Además del reducido número de especialistas en esta rama y la limitada distribución geográfica de los cardiólogos en el país, la detección y tratamiento de este tipo de enfermedades conlleva grandes costos tanto económicos como de tiempo y talento humano. Por tal motivo, se desea realizar la implementación de la inferencia de una red neuronal convolucional para detectar arritmias cardíacas en el microcontrolador Teensy 4.1, el cual posee características relevantes en temas de portabilidad, consumo energético y memoria. La red neuronal es la red SqueezeECG, esta red detecta arritmias cardíacas a partir de señales ECG sin preprocesamiento. La red ofrece una exactitud que está por encima de la de un cardiólogo en promedio. Un aspecto importante para resaltar en esta red es que tiene alrededor de 20x menos parámetros que modelos con una exactitud similar. Nuestros resultados muestran valores de accuracy, precisión, recall y F1 score por encima de 0.8. Se logró un tiempo de inferencia de 4205ms y una latencia de 8196ms trabajando a una frecuencia de 396MHz. El proceso de inferencia requiere un total de 813KB de memoria Flash y 750KB de memoria RAM. El código correspondiente a la implementación se encuentra disponible en el siguiente enlace: [https://github.com/JulianJerez/ECG\\_Microcontroller\\_Implementation](https://github.com/JulianJerez/ECG_Microcontroller_Implementation).

---

\*Trabajo de grado

\*\* Facultad de Ingeniería Físico-Mecánicas. Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Director: Carlos Arturo Boada Quijano, Máster en Ingeniería Electrónica. Codirector: Said David Pertuz Arroyo, PhD en Ciencias de la Computación.

## ABSTRACT

**TITLE:** A MICROCONTROLLER IMPLEMENTATION OF A CNN INFERENCE FOR DETECTING CARDIAC ARRHYTHMIAS\*

**AUTHOR:** JOSE LUIS CADENA TÉLLEZ, JULIAN EDUARDO JEREZ VERA\*\*

**KEY WORDS:** CARDIAC ARRHYTHMIAS, CONVOLUTIONAL NEURAL NETWORKS, DEEP LEARNING, INFERENCE AT THE EDGE, ELECTROCARDIOGRAPHIC SIGNALS, MICROCONTROLLER.

**DESCRIPTION:** Heart diseases have one of the highest mortality rates in Colombia. In addition, the small number of specialists in this field and their limited geographical distribution make the detection and treatment of this type of disease entail high costs in terms of time, money, and human talent. Therefore, we implemented the inference of a convolutional neural network to detect cardiac arrhythmias in the Teensy 4.1 microcontroller. This device has relevant characteristics in terms of portability, energy consumption, and memory. The neural network to implement is the SqueezeECG, which detects cardiac arrhythmias from ECG signals without preprocessing. It offers higher accuracy than the average cardiologist and has about 20 times fewer parameters than models of similar accuracy. Our results show accuracy, precision, recall, and F1 score values above 0.8. We achieved an inference time of 4205ms and latency of 8196ms at an operating frequency of 396MHz. The inference process required 813KB of Memory Flash and 750KB of Memory RAM. Code for our implementation is available at [https://github.com/JulianJerez/ECG\\_Microcontroller\\_Implementation](https://github.com/JulianJerez/ECG_Microcontroller_Implementation).

---

\* Degree work

\*\* Faculty of Physical – Mechanical Engineering. School of Electrical, Electronic and Telecommunications Engineering. Director: Carlos Arturo Boada Quijano, Master in Electronic Engineering. Co-director: Said David Pertuz Arroyo, PhD in Computer Science.

## INTRODUCCIÓN

Las arritmias cardíacas son un tipo de afección vascular que poseen una alta tasa de accidentes cerebrovasculares y muertes en todo el mundo cuando se detectan tardíamente. La detección temprana de las enfermedades cardíacas permite un tratamiento temprano y específico del paciente por parte del especialista, mejorando así su calidad de vida. Un método básico para adquirir información sobre el comportamiento eléctrico del corazón es el electrocardiograma. El ECG es la representación visual del voltaje en función del tiempo de la actividad eléctrica del corazón<sup>1</sup>, que se obtiene a través de sensores (electrodos) en el cuerpo del paciente, normalmente en el pecho, que captan las variaciones de potencial emitidas por el corazón. Esto permite al cardiólogo estudiar las patologías cardíacas obteniendo información sobre el comportamiento y la salud del corazón.

La extracción de las características más relevantes de las señales es la clave para la correcta detección de anomalías. Los métodos propuestos para la detección de arritmias cardíacas utilizan diferentes técnicas. Varios métodos utilizan Aprendizaje automático y Aprendizaje Profundo. El aprendizaje automático tradicional consta de dos etapas principales: Extracción/selección de características y clasificación. Para la etapa de clasificación se utilizan redes neuronales, máquinas de vectores de apoyo y análisis discriminante lineal. Los algoritmos de aprendizaje profundo realizan la extracción/selección de características y la clasificación dentro de la red profunda. Se han propuesto diversas arquitecturas de aprendizaje profundo, como las redes neuronales convolucionales (CNN), las redes neuronales recurrentes (RNN), las redes de creencia profunda (DBN) y el autocodificador<sup>2</sup>.

---

<sup>1</sup> LILLY, Leonard S. Pathophysiology of heart disease: a collaborative project of medical students and faculty. 6 ed. Lippincott Williams & Wilkins, 2012. ISBN 978-1451192759.

<sup>2</sup> PARVANEH, Saman, et al. Cardiac arrhythmia detection using deep learning: A review. *Journal of electrocardiology*. 2019, vol. 57, p. S70-S74.

La implementación de métodos de detección de arritmias en dispositivos embebidos es otro de los temas tratados en la literatura. Se propuso un dispositivo médico móvil (HeartSaver) para la autodetección de tres tipos de patologías cardíacas mediante algoritmos de detección<sup>3</sup>. Estos algoritmos pueden implementarse en dispositivos embebidos dada su sencillez y rendimiento en la detección de patologías cardíacas relevantes. De igual forma, se propuso una plataforma inalámbrica que recibe y muestrea la señal de ECG antes de enviarla por WIFI a un servidor informático<sup>4</sup>. Para la clasificación de arritmias, utilizaron redes neuronales residuales pre-entrenadas con una exactitud del 98,5%. También fue implementada una CNN de varias derivaciones para la detección de infarto de miocardio utilizando Matlab en un PC con Windows y una plataforma micro embebida.<sup>5</sup>. En primer lugar, diseñaron un algoritmo de segmentación de latidos para el preprocesamiento de los datos del ECG, con el fin de obtener latidos de varias derivaciones para capturar las principales características de cada una de ellas. Estas derivaciones son la entrada de la ML-CNN entrenada en MATLAB y que fue adaptada a la plataforma micro embebida sin entrenamiento extra, debido a la baja capacidad computacional del procesador que utilizaron. Los investigadores reportaron una sensibilidad del 95,40%, una especificidad del 97,37% y una exactitud del 96,00%. Además, los tiempos medios de procesamiento de cada latido son de aproximadamente 17,10 y 26,75 ms, para las plataformas MATLAB y ARM Cortex-A9, respectivamente.

---

<sup>3</sup> SANKARI, Ziad y ADELI, Hojjat. HeartSaver: A mobile cardiac monitoring system for auto-detection of atrial fibrillation, myocardial infarction, and atrio-ventricular block. *Computers in biology and medicine*. 2011, vol. 41, no 4, p. 211-220.

<sup>4</sup> CEN, Panida, et al. Intelligence ECG Monitoring System: Wireless platform and Arrhythmia classification using Residual neural network. *2019 12th Biomedical Engineering International Conference (BMEiCON)*. IEEE, 2019. p. 1-5.

<sup>5</sup> LIU, Wenhan, et al. Real-time multilead convolutional neural network for myocardial infarction detection. *IEEE journal of biomedical and health informatics*. 2017, vol. 22, no 5, p. 1434-1444.

Asimismo, la implementación de la inferencia de una red neuronal en un microcontrolador de baja resolución fue propuesta, la cual utiliza métodos de cuantificación para reducir los requisitos de recursos<sup>6</sup>. Sus resultados muestran una exactitud del 94% para 22 bits. En otro estudio, se implementó la inferencia de una red neuronal convolucional-recurrente para la detección de arritmias cardíacas, en un System-on-Chip nRF52 de bajo consumo con un núcleo de procesamiento Cortex- M4 de ARM<sup>7</sup>. Utilizaron un proceso de precisión de 8 bits en punto fijo para implementar el modelo y reducir el consumo de memoria. La cuantificación produce una disminución en el F1 score de 0,8 a 0,784 respecto a la implementación original. También señalaron que el requerimiento de memoria era de 195,5 KB y el rendimiento de 33,98 MOps/s.

En este estudio, aspiramos a implementar la inferencia de un modelo modificado del Tiny Model<sup>8</sup> propuesto en un microcontrolador. Para llevar a cabo este proceso, creamos funciones que emulan el comportamiento de las capas que componen la red. La ventaja de estas funciones es la versatilidad para ser utilizadas en otros modelos ya que sólo dependen de los parámetros definidos en la descripción de la CNN. Nuestro trabajo se centra en realizar la inferencia sin utilizar móviles, dispositivos WIFI u ordenadores.

---

<sup>6</sup> JARAMILLO-RUEDA, Andrés F.; VARGAS-PACHECO, Laura Y y FAJARDO, Carlos A. A Computational Architecture for Inference of a Quantized-CNN for Detecting Atrial Fibrillation. *Ingeniería y Ciencias*. 2020, vol. 16, no 32, p. 135-149.

<sup>7</sup> FARAONE, Antonino y DELGADO-GONZALO, Ricard. Convolutional-recurrent neural networks on low-power wearable platforms for cardiac arrhythmia detection. *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2020. p. 153-157.

<sup>8</sup> FAJARDO, Carlos y ROY, Kaushik. A Compressed Deep Learning Model to detect Cardiac Arrhythmias. *43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2021. p.1–4.

## 1. DATA SET

Para nuestro proyecto utilizamos el Icentia11K<sup>9</sup>, un data set ECG compuesto por señales obtenidas de 11.000 pacientes. Estas señales fueron tomadas de pacientes principalmente situados en Ontario, Canadá, con un promedio de edad de  $62.2 \pm 17.4$  años.

Las señales crudas se obtuvieron usando un dispositivo de monitorización cardiaca de una sola derivación, el CartioSTAT<sup>TM</sup><sup>10</sup>, y fueron registradas con una resolución de 16-bits y muestreadas a una frecuencia de 250 Hz. Cada paciente utilizó el dispositivo durante alrededor de una semana. Estos registros fueron segmentados en segmentos de 70 minutos de muestras de señal. Para reducir el tamaño del data set, se seleccionó un porcentaje aleatorio de estos segmentos, que finalmente fueron segmentados en tramos de 2049 muestras. Estos tramos corresponden a las señales del data set, el cual cuenta con 3 etiquetas de ritmo que corresponde al tipo de señal: NSR (Ritmo Normal), AFib (Fibrilación auricular), and AFlutter (Flutter auricular)<sup>11</sup>.

---

<sup>9</sup> TAN, Shawn, *et al.* Icentia11k: An unsupervised representation learning dataset for arrhythmia subtype discovery. *arXiv preprint*. 2019, p. 1–10, arXiv:1910.09570.

<sup>10</sup> PAQUET, Pierre; LEVESQUE, David y FECTEAU, Pierre. *Adhesive extender for medical electrode and use thereof with wearable monitor*. U.S. Patent Application No 16/093,151, 20 Jun. 2019.

<sup>11</sup> TAN, S. Op. cit, p.1-10

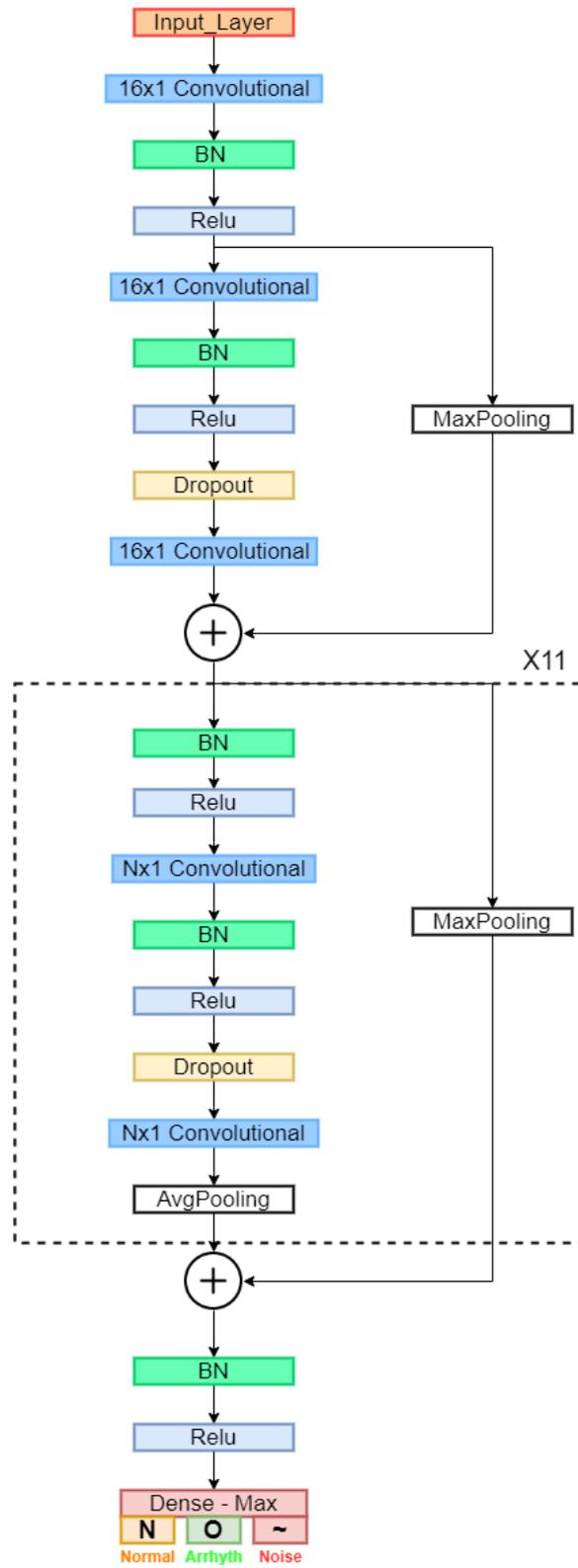
## 2. RED NEURONAL CONVOLUCIONAL

El modelo implementado en este trabajo se basó en el Tiny Model de la SqueezeECG<sup>12</sup>, mostrado en la figura 1. La red neuronal tiene una capa de entrada que recibe las señales ECG crudas. Esta red está compuesta por 11 bloques residuales. Para evitar la saturación y prevenir el overfitting, se utilizaron capas tipo Batch y capas Dropout, respectivamente. Estas capas son seguidas por una capa ReLU. Además, se utilizan capas convolucionales. Para estas capas, se inicia con 16 filtros, los cuales aumentan a 32 filtros a partir del noveno bloque residual. También se usan capas tipo Pooling: Maxpooling y Avgpooling. Todos los bloques residuales cuentan con un stride de 1, a excepción del primero y noveno bloque, cuyo stride es de 2. Finalmente, la capa de salida es una capa Fully Connected seguida de una función máxima creada por nosotros, la cual cuenta con tres tipos de salidas (Normal, Arritmia y Ruido). La función máxima le asigna un valor de 1 a la salida de la capa Fully Connected de mayor valor, la cual corresponderá al tipo de señal que entró a la red neuronal, y al resto de las salidas se le asigna un valor de 0.

---

<sup>12</sup>FAJARDO, Carlos y ROY, Kaushik. A Compressed Deep Learning Model to detect Cardiac Arrhythmias. *43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2021. p.1–4.

Figura 1. Modelo implementado. Modificación del Tiny Model.



### 3. IMPLEMENTACIÓN EN EL MICROCONTROLADOR

El modelo fue implementado en el microcontrolador Teensy 4.1, el cual está basado en el ARM Cortex-M7 y trabaja hasta a 600 MHz. Cuenta con una memoria FLASH de 8MB y una memoria RAM de 1MB. La RAM está dividida en dos partes iguales: SRAM(RAM1) y PSRAM(RAM2)<sup>13</sup>. Para obtener la señal cardíaca utilizamos un kit ECG con electrodos basado en la tarjeta AD8232<sup>14</sup>. Para la implementación del modelo se utilizó el lenguaje de descripción C en el IDE de Arduino.

La red neuronal convolucional posee un total de 191251 parámetros, que representan un consumo de memoria de 765KB utilizando variables de tipo float de 32-bits para el almacenamiento de los parámetros de la red.

#### 3.1. ADQUISICIÓN DE LA SEÑAL DE ENTRADA

El modelo fue entrenado utilizando 4000 señales de pacientes del data set Icentia11K<sup>15</sup>. Debido a las especificaciones de las señales del data set, la señal de entrada adquirida por el sensor de señales ECG es normalizada y muestreada a 250Hz.

La señal cruda de entrada obtenida por el sensor, mostrada en la ecuación 1, es muestreada por medio de interrupciones usando los timers del microcontrolador,

---

<sup>13</sup> SPARKFUN ELECTRONICS. [Sitio Web]. SPARKFUN, Teensy 4.1. [Consulta: 6 de abril 2021]. Disponible en: <https://www.sparkfun.com/products/16771>

<sup>14</sup> .SPARKFUN ELECTRONICS. [Sitio Web]. SPARKFUN, SparkFun Single Lead Heart Rate Monitor - AD8232. [Consulta: 15 de abril 2021]. Disponible en: <https://www.sparkfun.com/products/12650>.

<sup>15</sup>TAN, Shawn, *et al.* Icentia11k: An unsupervised representation learning dataset for arrhythmia subtype discovery. *arXiv preprint*. 2019, p. 1–10, arXiv:1910.09570.

mientras se está ejecutando la inferencia de la red. Usamos la función IntervalTimer<sup>16</sup> compatible con Arduino, la cual llama a una función en un intervalo de tiempo dado. En este caso, la interrupción ocurre cada 4ms con el fin de cumplir la frecuencia de muestreo del set de entrenamiento.

Consideremos  $x$  la señal cruda de entrada:

$$x_i = [x_{i1}, x_{i2}, \dots, x_{iN-1}, x_{iN}] \quad (1)$$

Donde  $N$  es el número de muestras, en este caso, 2049.

Debido a que la inferencia de la red tarda menos tiempo que el muestreo de la señal de entrada, tan pronto se finaliza la inferencia, se ejecuta una función de espera con el fin de asegurar las 2049 muestras necesarias para iniciar un nuevo proceso de inferencia. Una vez el proceso de muestreo ha finalizado, la señal es normalizada utilizando la ecuación 2.

$$\bar{x}_i = \frac{x_i - \mu}{\sigma} \quad (2)$$

Donde  $\mu$  es la media de la señal cruda de entrada, calculada usando la ecuación 3.

$$\mu = \frac{\sum_{n=1}^N x_{in}}{N} \quad (3)$$

$\sigma$  es la desviación estándar, calculada usando la ecuación 4.

---

<sup>16</sup>PJRC. [Sitio Web]. PJRC, IntervalTimer. [Consulta: 25 de junio 2021]. Disponible en: [https://www.pjrc.com/teensy/td\\_timing\\_IntervalTimer.html](https://www.pjrc.com/teensy/td_timing_IntervalTimer.html).

$$\sigma = \sqrt{\frac{\sum_{n=1}^N (x_{in} - \mu)^2}{N}} \quad (4)$$

Para determinar el valor de la media y la desviación estándar usamos las señales del data set. Estos valores son 503.06 y 109.83, respectivamente.

### 3.2. FUNCIONES DE ARDUINO

La implementación de la inferencia de la red neuronal en el microcontrolador utilizando Arduino es un desafío en términos de la representación de las capas en este lenguaje de programación. Por esta razón, creamos funciones que emulan el comportamiento de cada una de las capas del modelo.

Se creó un archivo tipo header donde se definieron funciones parametrizadas de Arduino para cada una de las capas, con sus correspondientes operaciones matemáticas, basándonos en la operación interna de cada una de las capas encontrada en la documentación oficial de Keras<sup>17</sup>, favoreciendo la implementación de la CNN completa. Además, creamos dos nuevas funciones: *Create\_Matrix* y *Clear\_Matrix*. La primera utiliza la función Malloc para asignar un espacio de memoria a una variable en la RAM2. La otra función me permite hacer una limpieza de una variable (asignar todas las posiciones de la variable igual a cero), con el fin de evitar valores indeseados que pueden aparecer al reutilizar una variable dentro del mismo código de Arduino. Estas dos funciones nos permitieron utilizar solamente dos variables globales para toda la implementación de la red neuronal, permitiendo reducir el consumo de memoria RAM.

La figura 2 muestra el flujograma de los dos procesos principales, la inferencia de la CNN y la adquisición de la señal de entrada. La duración de estos procesos a través del tiempo se muestra en la figura 3. Debido a que la adquisición de la señal

---

<sup>17</sup>KERAS. [Sitio Web]. KERAS, Keras layers API. [Consulta: 10 de mayo 2021]. Disponible en: <https://keras.io/api/layers/>.

de entrada se realiza por medio de interrupciones, existen lapsos donde ambos procesos están ocurriendo al mismo tiempo.

Figura 2. Flujograma del modelo.

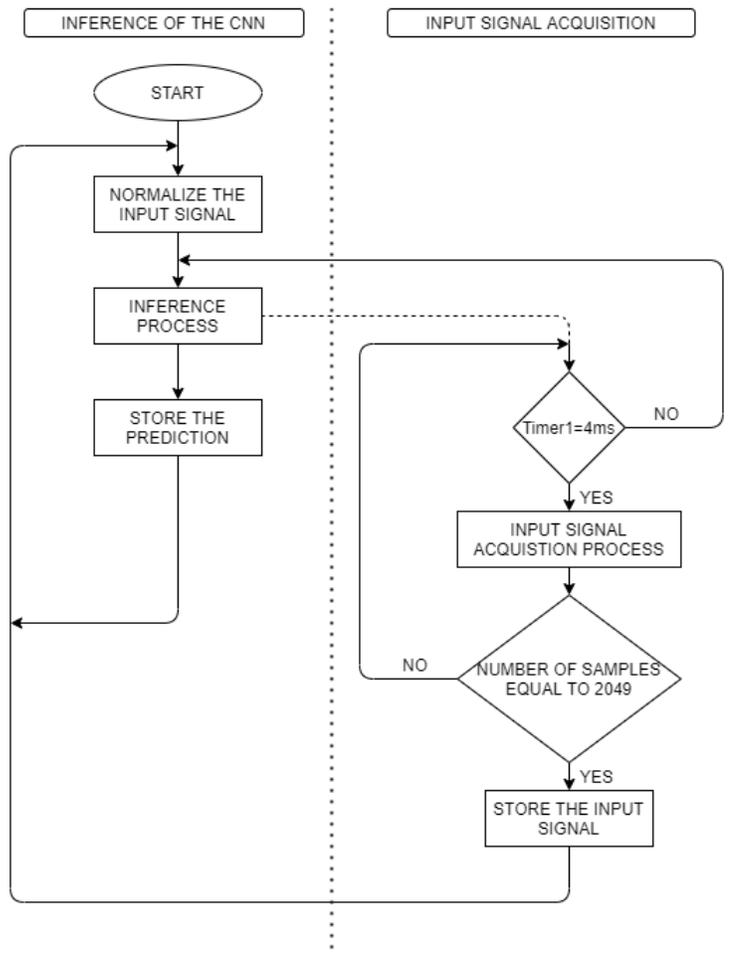
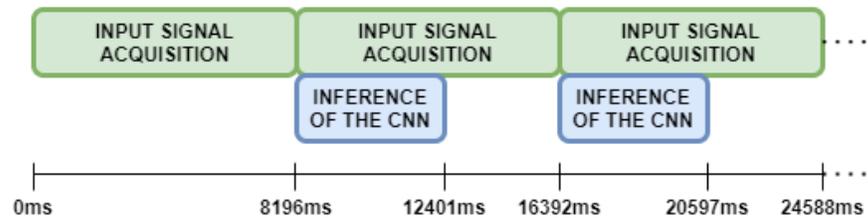


Figura 3. Procesos ejecutados en el microcontrolador.



#### 4. RESULTADOS

La tabla 1 muestra la matriz de confusión del modelo implementado. Esta matriz muestra los porcentajes de las señales normal, arritmia o ruido que fueron correctamente e incorrectamente clasificadas. Por ejemplo, el 90.7% corresponde al porcentaje de señales normales que fueron predichas correctamente como señales normales, el 1.1% corresponde al porcentaje de señales normales que fueron predichas incorrectamente como señal arritmia, etc.

Tabla 1. Matriz de confusión del modelo.

	<b>Predicción Normal</b>	<b>Predicción Arritmia</b>	<b>Predicción Ruido</b>
<b>Señal Normal</b>	90.7%	1.1%	8.1%
<b>Señal Arritmia</b>	25.1%	65.2%	9.6%
<b>Señal Ruido</b>	10.9%	1.4%	87.6%

Las medidas para evaluar el rendimiento de la red implementada son: Exactitud, precisión, recall y F1 score (Tabla 2). La exactitud es el porcentaje total de señales que fueron clasificadas correctamente por el modelo. La precisión es el porcentaje de los verdaderos positivos de todas las predicciones positivas. El recall es el porcentaje de casos que son identificados correctamente. El F1 score es la media armónica de la precisión y el recall.

Tabla 2. Medidas de rendimiento del modelo.

<b>Medidas</b>	
Medida	Valor
Exactitud	0.882
Precisión	0.812
Recall	0.812
F1 Score	0.811

La tabla 3 muestra la memoria del microcontrolador utilizada para la implementación de la red neuronal. Los parámetros de la red fueron almacenados en la memoria flash utilizando la función PROGMEM<sup>18</sup> ya que estos valores no cambian durante la inferencia. Los valores intermedios fueron almacenados en la memoria SRAM y PSRAM.

Tabla 3. Memoria utilizada en la implementación del modelo.

<b>Teensy 4.1</b>			
Memoria	Disponible	Usada	%Usada
Flash	8MB	813kB	10%
RAM1	512kB	357kB	68%
RAM2	512kB	393kB	77%

Se puede observar que en la tabla 2 todas las medidas están por encima de 0.8, haciendo de este un modelo confiable. La tabla 3 muestra que el consumo de memoria es inferior a 1MB para la implementación de la red neuronal, debido al número de parámetros que esta posee.

<sup>18</sup> ARDUINO. [Sitio Web]. ARDUINO, Progmem. [Consulta: 13 de junio 2021]. Disponible en: <https://www.arduino.cc/reference/tr/language/variables/utilities/progmem/>.

Tabla 4. Características de desempeño del microcontrolador.

**Características de desempeño**

Frecuencia de reloj (CF)	Tiempo de inferencia
150 [MHz]	10922 [ms]
<b>396 [MHz]</b>	<b>4205 [ms]</b>
450 [MHz]	3702 [ms]
528 [MHz]	3182 [ms]
600 [MHz]	2802 [ms]

Como se muestra en la figura 3, tenemos una latencia de 8196ms. Para garantizar una correcta predicción, es necesario un tiempo inferior para el proceso de inferencia de la red. La tabla 4 resume el tiempo que tarda en realizarse la inferencia de la red para cada una de las frecuencias a las que el microcontrolador es capaz de trabajar. Por esta razón, utilizamos una frecuencia de trabajo de 396MHz por la implementación de la red. Esta frecuencia es la menor que cumple con los requisitos de tiempo, lo cual nos permite reducir el consumo de potencia del microcontrolador.

## 5. CONCLUSIONES

Presentamos la implementación de una modificación del modelo Tiny Model de la SqueezeECG<sup>19</sup> en un microcontrolador de 32-bits (Teensy 4.1). Desarrollamos funciones parametrizadas en C que describen el comportamiento de la capas utilizadas en el modelo, las cuales son una contribución para la implementación de modelos de aprendizaje profundo en sistemas embebidos. La implementación logro una exactitud de 0.882 utilizando menos del 15% de memoria FLASH disponible.

El trabajo futuro estará enfocado en la creación de nuevas funciones para capas que no fueron utilizadas en este modelo, y mejorar las funciones utilizadas en este trabajo para la implementación de modelos más complejos en sistemas embebidos.

---

<sup>19</sup>FAJARDO, Carlos y ROY, Kaushik. A Compressed Deep Learning Model to detect Cardiac Arrhythmias. *43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2021. p.1–4.

## BIBLIOGRAFIA

ARDUINO. [Sitio Web]. ARDUINO, Progmem. [Consulta: 13 de junio 2021].  
Disponible en:  
<https://www.arduino.cc/reference/tr/language/variables/utilities/progmem/>.

CEN, Panida, *et al.* Intelligence ECG Monitoring System: Wireless platform and Arrhythmia classification using Residual neural network. *2019 12th Biomedical Engineering International Conference (BMEiCON)*. IEEE, 2019. p. 1-5.

FAJARDO, Carlos y ROY, Kaushik. A Compressed Deep Learning Model to detect Cardiac Arrhythmias. *43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2021. p.1–4.

FARAONE, Antonino y DELGADO-GONZALO, Ricard. Convolutional-recurrent neural networks on low-power wearable platforms for cardiac arrhythmia detection. *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2020. p. 153-157.

HASAN, Md Abid, *et al.* Cardiac Arrhythmia Detection in an ECG Beat Signal Using 1D Convolution Neural Network. *2020 IEEE Region 10 Symposium (TENSYP)*. IEEE, 2020. p. 352-357.

JARAMILLO-RUEDA, Andrés F.; VARGAS-PACHECO, Laura Y y FAJARDO, Carlos A. A Computational Architecture for Inference of a Quantized-CNN for Detecting Atrial Fibrillation. *Ingeniería y Ciencias*. 2020, vol. 16, no 32, p. 135-149.

KERAS. [Sitio Web]. KERAS, Keras layers API. [Consulta: 10 de mayo 2021]. Disponible en: <https://keras.io/api/layers/>.

LILLY, Leonard S. Pathophysiology of heart disease: a collaborative project of medical students and faculty. 6 ed. Lippincott Williams & Wilkins, 2012. ISBN 978-1451192759.

LIU, Wenhan, *et al.* Real-time multilead convolutional neural network for myocardial infarction detection. *IEEE journal of biomedical and health informatics*. 2017, vol. 22, no 5, p. 1434-1444.

MOCERINO, Luca y CALIMERA, Andrea. Fast and accurate inference on microcontrollers with boosted cooperative convolutional neural networks (bc-net). *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2020, vol. 68, no 1, p. 77-88.

PARVANEH, Saman, *et al.* Cardiac arrhythmia detection using deep learning: A review. *Journal of electrocardiology*. 2019, vol. 57, p. S70-S74.

PAQUET, Pierre; LEVESQUE, David y FECTEAU, Pierre. *Adhesive extender for medical electrode and use thereof with wearable monitor*. U.S. Patent Application No 16/093,151, 20 Jun. 2019.

PJRC. [Sitio Web]. PJRC, IntervalTimer. [Consulta: 25 de junio 2021]. Disponible en: [https://www.pjrc.com/teensy/td\\_timing\\_IntervalTimer.html](https://www.pjrc.com/teensy/td_timing_IntervalTimer.html).

POURBABAEE, Bahareh; ROSHTKHARI, Mehrsan Javan y KHORASANI, Khashayar. Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2018, vol. 48, no 12, p. 2095-2104.

SANKARI, Ziad y ADELI, Hojjat. HeartSaver: A mobile cardiac monitoring system for auto-detection of atrial fibrillation, myocardial infarction, and atrio-ventricular block. *Computers in biology and medicine*. 2011, vol. 41, no 4, p. 211-220.

SPARKFUN ELECTRONICS. [Sitio Web]. SPARKFUN, SparkFun Single Lead Heart Rate Monitor - AD8232. [Consulta: 15 de abril 2021]. Disponible en: <https://www.sparkfun.com/products/12650>.

SPARKFUN ELECTRONICS. [Sitio Web]. SPARKFUN, Teensy 4.1. [Consulta: 6 de abril 2021]. Disponible en: <https://www.sparkfun.com/products/16771>.

TAN, Shawn, *et al.* Icentia11k: An unsupervised representation learning dataset for arrhythmia subtype discovery. *arXiv preprint*. 2019, p. 1–10, arXiv:1910.09570.

ZHANG, Wenbo, *et al.* ECG signal classification with deep learning for heart disease identification. *2018 International Conference on Big Data and Artificial Intelligence (BDAI)*. IEEE, 2018. p. 47-51.