

Sistema de Monitorización no Intrusiva de Potencia en una Red Eléctrica Doméstica empleando
una Raspberry Pi y una Aplicación Web

Laura Sofía Mendoza Díaz

Viviana Lisbeth Espitia Mantilla

Trabajo de Grado para Optar al Título de Ingeniero Electrónico

Directora

Yulieth Jiménez Manjarres, Dra. en Ingeniería Electrónica

Codirectores

César Antonio Duarte Gualdrón, PhD. Electrical and Computer Engineering

Carlos Andrés Angulo Julio, Magister en Ingeniería Electrónica

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2022

Dedicatoria

A mis padres, Celina y Enrique que desde pequeña me inculcaron el amor por el aprendizaje, a mis hermanas, Mafer y Diana por darme la fuerza necesaria para seguir luchando, a mi familia, este logro es todo nuestro.

A mis mejores amigos, Santiago y Yoseth, mi familia escogida, nada de esto sería posible sin su apoyo y acompañamiento constante; a Silvia, aunque no sé por donde nos lleve la vida de aquí en adelante, es parte de mí, de la manera en que veo el mundo y de todo lo que soy hoy, la llevo conmigo a donde sea que vaya; a Viviana, no pude haber tenido mejor compañera para este camino, crecimos juntas estos 7 años y llegó la hora de empezar a construir sobre toda la base que hemos forjado; a Maleja, tú sabes porque; a todos los compañeros y compañeras que me deja la universidad, tengo la fortuna de tener alrededor, gente super inteligente, amable, colaboradora, siempre dispuestos a aprender juntos, son parte importante de este viaje. A Taylor Swift por darme la luz que necesitaba en el momento más oscuro de mi vida y continuar acompañándome con sus letras en cada uno de los pasos que doy; a Miley Cyrus, Demi Lovato y Martin Cirio, admiración total para el trabajo que hacen y que me acompaña en cada aventura que decido emprender.

Al amor de mi vida, que aún sin conocer lleva consigo parte de mi.

-Laura Mendoza

A mi madre Nelly, sin su apoyo, comprensión y amor no habría llegado hasta este punto de mi vida, a mi padre Carlos, por su acompañamiento, a mis hermanos, Carlos y Sebastian, por estar pendientes de mí en todo este proceso.

A mis compañeros de universidad con quienes pasamos grandes momentos, por ser personas únicas, dispuestos a compartir su conocimiento y brindar ayuda, siempre serán una parte de mí.

Y por último pero no menos importante al amor de mi vida Luis, por brindarme su apoyo incondicional en cada momento de dificultad y creer en mí.

A todos ellos, gracias.

-Viviana Espitia

Agradecimientos

A la profesora Yulieth Jiménez por guiarnos y acompañarnos continuamente en todo este proceso.

A los profesores César Gualdrón y Carlos Angulo por la colaboración y asesoría, vital para el desarrollo de este proyecto.

A todas las personas que intervinieron de manera directa o indirecta en la realización de este trabajo, gracias totales.

Tabla de Contenido

Introducción	13
1. Marco teórico	17
1.1. Monitorización de carga	17
1.2. Bases de datos	18
1.3. Machine Learning	19
1.3.1. Deep Neural Networks	19
1.3.2. Support Vector Machine	20
1.3.3. Validación cruzada	21
1.3.4. Evaluación de modelos de clasificación	22
1.4. Raspberry Pi 3 B+	26
1.5. Angular CLI	27
1.6. Google Firebase	28
2. Base de datos	29
2.1. Electrodomésticos	30
2.2. Mediciones	30
2.3. Mediciones individuales	31
2.4. Tensión - Corriente	31

3. Identificación de electrodomésticos	33
3.1. Algoritmos en ambiente de prueba	33
3.2. Datos de prueba	33
3.3. Ejecución en ambiente de prueba	34
3.4. Ejecución en el sistema destino	39
4. Aplicativo Web	47
4.1. Base de datos	47
4.2. Cálculo de la potencia	48
4.3. Visualización	49
5. Recomendaciones	53
6. Conclusiones	54
Referencias Bibliográficas	56
Apéndices	57

Lista de Figuras

Figura 1.	Arquitectura de una DNN.	20
Figura 2.	Algoritmo SVM.	22
Figura 3.	Esquema de la validación cruzada.	23
Figura 4.	Estructura de una Matriz de Confusión 2x2	24
Figura 5.	Representación de la ROC-AUC	26
Figura 6.	Pines Raspberry Pi 3 B+	27
Figura 7.	Arquitectura de la base de datos relacional	29
Figura 8.	Muestra de la tabla 1_Appliances	30
Figura 9.	Muestra de la tabla 2_Measurements	31
Figura 10.	Muestra de la tabla 3_Individual_Measurement	32
Figura 11.	Muestra de la tabla 4_MI_0_1	32
Figura 12.	Matrices de confusión para el primer escenario	36
Figura 13.	Curvas ROC para el primer escenario	37
Figura 14.	Matrices de confusión para el segundo escenario	38
Figura 15.	Curvas ROC para el segundo escenario	39
Figura 16.	Ejecución del algoritmo en la Raspberry para el ventilador	41
Figura 17.	Ejecución del algoritmo en la Raspberry para el secador de cabello	42

Figura 18.	Ejecución del algoritmo en la Raspberry para el computador	43
Figura 19.	Matriz de confusión del modelo SVM en la Raspberry	44
Figura 20.	Curvas ROC del modelo SVM en la Raspberry	45
Figura 21.	Estructura de las colecciones.	48
Figura 22.	Tensión y corriente de un dispositivo CFL (id = 55) acotados en un periodo.	49
Figura 23.	Pantalla de login	50
Figura 24.	Sección Electrodomésticos con cuatro dispositivos conectados.	51
Figura 25.	Sección Electrodomésticos con cuatro dispositivos desconectados.	51
Figura 26.	Sección Potencia Total.	52

Lista de Tablas

Tabla 1.	Base de Datos PLAID	34
Tabla 2.	Base de Datos GISEL	35
Tabla 3.	Resultados de los modelos de clasificación en el primer escenario	37
Tabla 4.	Resultados de los modelos de clasificación en el segundo escenario	39
Tabla 5.	Resultados del modelo de clasificación SVM en la Raspberry	46
Tabla 6.	Estructura de la base de datos del aplicativo web	47

Lista de Apéndices

	pág.
Apéndice A. Código principal ejecutado en la Raspberry	57

Resumen

Título: Sistema de monitorización no intrusiva de potencia en un red eléctrica doméstica empleando una Raspberry Pi y una aplicación web *

Autoras: Viviana Lisbeth Espitia Mantilla, Laura Sofía Mendoza Díaz **

Palabras Claves: Desagregación de carga, Raspberry Pi, identificación de carga, monitor no intrusivo, aplicación web, inteligencia artificial, máquinas de soporte vectorial.

Descripción: Un sistema de monitorización no intrusivo de carga mide la tensión y la corriente directamente de la red y utiliza técnicas de desagregación para identificar los electrodomésticos conectados. En este trabajo de grado se implementan algoritmos de máquinas de soporte vectorial (SVM) y de redes neuronales profundas (DNN) para la desagregación de carga, y se compara su desempeño. Para ello, se crea una base de datos con las señales de varios electrodomésticos en MySQL, las cuales son caracterizadas y enviadas al modelo de clasificación para predecir cuál es el dispositivo encendido. Además, la página web es actualizada con la información del dispositivo detectado, cuya visualización incluye el estado de cada elemento, la potencia instantánea y la potencia total consumida. La creación de la base de datos permite un mejor control y óptima accesibilidad de la información. Por otra parte, la comparación de algoritmos presenta un resultado destacado del modelo SVM con un 15 % de exactitud mayor en contraste al DNN. Así mismo, su implementación en la tarjeta Raspberry Pi brinda una aproximación más real, abarcando la clasificación de 14 diferentes dispositivos incluidos en la base de datos, enlazando estos aparatos detectados directamente al aplicativo web en donde se presenta la información adquirida aportando un valor agregado al usuario final.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Directora: Yulieth Jiménez Manjarres, Dra. en Ingeniería Electrónica. Codirectores: César Antonio Duarte Gualdrón, PhD. Electrical and Computer Engineering y Carlos Andrés Angulo Julio, Magister en Ingeniería Electrónica

Abstract

Title: Non intrusive power monitoring system for a home power grid using a Raspberry Pi and a web application *

Authors: Viviana Lisbeth Espitia Mantilla, Laura Sofía Mendoza Díaz **

Keywords: Load disaggregation, Raspberry Pi, load identification, non intrusive monitor, web application, artificial intelligence, support vector machine.

Description: A non-intrusive load monitoring system measures voltage and current directly from the grid and uses dis-aggregation techniques to identify connected appliances. In this undergrad work, support vector machines (SVM) and deep neural network (DNN) algorithms are implemented for load disaggregation, and their performance is compared. For this purpose, a database is created with signals from various home appliances in MySQL, which are characterized and sent to the classification model to predict which device is turned on. In addition, the web page is updated with the information of the detected device, whose visualization includes the status of each element, the instantaneous power and the total power consumed. The creation of the database allows better control and optimal accessibility of the information. On the other hand, the comparison of algorithms shows an outstanding result of the SVM model with a 15% higher accuracy in contrast to the DNN. Likewise, its implementation in the Raspberry Pi board provides a more realistic approach, covering the classification of 14 different devices included in the database, linking these detected devices directly to the web application where the acquired information is presented, providing an added value to the end user.

* Bachelor Thesis

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Yulieth Jiménez Manjarres, Doctor in Electronic Engineering. Advisors: César Antonio Duarte Gualdrón, PhD. Electrical and Computer Engineering, and Carlos Andrés Angulo Julio, Master in Electronic Engineering

Introducción

La gran demanda de energía a nivel mundial se prevé que alcance los 33300 [TWh] para el 2030. En América Latina y el Caribe la demanda energética ha ido creciendo a una tasa del 3,2 % del 2000 al 2014 y se estima que aumente de 2,3 a 2,7 veces hasta 2060 (World-Energy-Council, 2017). Para suplir esta demanda se podría incrementar el suministro de energía, sin embargo, esto tendría fuertes impactos en el medio ambiente y en la economía. Por tal razón, es indispensable buscar otras soluciones más eficientes, ya que podría generarse una crisis de escasez de energía.

La eficiencia energética se presenta como una alternativa a este problema, la cual puede incluir brindar información detallada del consumo, lo que ha demostrado que podría motivar a los usuarios a disminuir su gasto energético entre un 5 % y 15 % (Darby, 2006). Por otro lado, el enfoque de desagregación de carga, que consiste en entregar cifras del consumo de cada electrodoméstico y dar a conocer al usuario cuál es el elemento que está consumiendo más energía en la vivienda, ha logrado una mayor concientización, ya que, según estudios realizados, el 81 % de los hogares analizados encontraron útil la información y un 38 % demostró haber aprendido algo nuevo de ello (Darby, 2006). La consecuencia de estos cambios fue un mayor ahorro de energía en las viviendas.

El objetivo de la monitorización de carga es obtener automáticamente información detallada identificando cada una de las firmas de carga generadas por los electrodomésticos del hogar

(Salgado y Ávila, 2021). Esta tarea puede realizarse de manera “intrusiva”, es decir, instalando un sensor en cada dispositivo, recopilando datos en un sistema centralizado y haciendo uso de plataformas de comunicación; o “no intrusiva”, que es tomando medidas en un único punto para registrar el funcionamiento de un grupo de aparatos y luego aplicar algoritmos matemáticos para hacer la desagregación (Jiménez y cols., 2014). Ambos sistemas de monitorización siguen un procedimiento estipulado en etapas de detección, procesamiento y visualización de los datos.

La monitorización no intrusiva de carga (NILM, por sus siglas en inglés) ha generado un creciente interés para estimar la operación y el consumo individual de potencia de los aparatos residenciales ya que reduce los esfuerzos de hardware asociados a los sistemas multisensor o monitoreo intrusivo mencionado anteriormente (Jiménez y cols., 2015). El proceso de identificación funciona analizando los cambios de corriente y tensión en señales transitorias o de estado estable, para extraer las características más importantes para la segregación de información (Salgado y Ávila, 2021).

NILM puede ser visto como un problema de identificación de clases múltiples que podría resolverse con técnicas de Machine Learning, por lo que el método de clasificación SVM podría resultar beneficioso para este propósito, ya que permite encontrar un hiperplano que separe de manera óptima los vectores de características o instancias en diferentes clases.

En el grupo GISEL de la Universidad Industrial de Santander se vienen desarrollando inves-

tigaciones referentes a la monitorización no intrusiva para desagregación de carga. Como producto de estas investigaciones en una Tesis Doctoral y un Trabajo de Investigación de maestría, se cuentan con algoritmos para la identificación de electrodomésticos y la estimación de la potencia que consumen.

El presente proyecto busca establecer la organización de la base de datos que se empleará en un sistema de monitorización no intrusiva en un dispositivo portable, realizar el procesamiento con algoritmos optimizados bajo la medición de diferentes métricas y comparar metodologías modernas de aprendizaje profundo tales como máquinas de soporte vectorial (SVM) y redes neuronales profundas (DNN), enlazados con una aplicación web en donde se puedan encontrar datos de consumo, para su visualización en tiempo real.

El objetivo general planteado en este trabajo es “Implementar algoritmos de desagregación de carga mediante máquinas de soporte vectorial (SVM) y de estimación de potencia, sobre una Raspberry Pi 3, a partir de señales eléctricas de electrodomésticos”. Además, para el cumplimiento de este objetivo se propusieron los siguientes objetivos específicos:

1. Elaborar una base de datos a partir de un conjunto de señales de electrodomésticos adquiridas previamente por el grupo GISEL
2. Evaluar el desempeño de la clasificación realizada por máquinas de soporte vectorial (SVM) en contraste con la de redes neuronales profundas (DNN).

3. Implementar en una Raspberry Pi3 algoritmos basados en máquinas de soporte vectorial (SVM) para la clasificación de señales eléctricas de electrodomésticos.
4. Visualizar los resultados de desagregación de carga y estimación de potencia en una página web.

Este trabajo está dividido en 6 capítulos, en los cuales se presenta inicialmente una contextualización del desarrollo del proyecto y las herramientas usadas. De igual forma, se exponen cada una de las secciones propuestas en los alcances, como lo son, la base de datos, el desarrollo del algoritmo para la identificación de elementos, la implementación en la Raspberry Pi 3 B+ y la actualización de la página web. Para finalizar se presentan las conclusiones y recomendaciones para trabajos futuros.

1. Marco teórico

En este capítulo se abordan los conceptos más importantes para el proyecto, comenzando por la definición de monitorización de la carga la cual es la base fundamental, siguiendo con la explicación de las bases de datos, sus características y metodologías de diseño más importantes. Además, se plantea el tema de aprendizaje automático en el cual se describen las arquitecturas a usar y los conceptos básicos significativos para entender cada una de las herramientas utilizadas, tanto para entrenar el modelo como para evaluar su desempeño. Así mismo, se exponen las principales características de la tarjeta utilizada para la implementación del algoritmo. Por último, se plantean conceptos básicos de la programación web como lo son Angular CLI y Google Firebase.

1.1. Monitorización de carga

La monitorización de carga es un campo de investigación cuyo objetivo es lograr una trazabilidad fiable de la carga eléctrica en un determinado espacio. Tradicionalmente, consiste en el uso de información histórica (lo que puede conducir a estimaciones inexactas) o sistemas con sensores dedicados para el seguimiento de las señales (que son más precisos pero aumentan los esfuerzos de hardware, los costes de instalación y el mantenimiento de los sensores y plataformas de comunicación). Sin embargo, con el auge de la inteligencia artificial, surgió un nuevo camino que minimiza los problemas anteriormente mencionados, la monitorización no intrusiva de carga o NILM por sus siglas en inglés. Un sistema NILM emplea mediciones en la caja de distribución eléctrica (directamente en la red) y utiliza técnicas de desagregación de carga automática para la identificación de los aparatos conectados. Este procedimiento disminuye notablemente los sensores utilizados y

brinda estimaciones más certeras dado el poder computacional de los algoritmos desarrollados.

1.2. Bases de datos

Una base de datos es un conjunto de datos interrelacionados almacenados en un lugar específico, ya sea virtual o físico. Cuando su almacenamiento es virtual, como será utilizado para este trabajo, lo más común es que se use para una aplicación, aunque siempre funcionará independientemente de los programas a los que se enlace. Dependiendo de la manera en que se organicen los datos se puede tener la siguiente clasificación:

- **Bases de datos relacionales:** Los elementos se representan como una matriz de tablas con filas y columnas, proporcionando de forma flexible y eficiente el acceso a la información. Este tipo de bases de datos se representan a través de modelos de entidad-vínculo. Las relaciones de estos suelen estar clasificadas en 3 grupos, 1:1 (de uno a uno), 1:N (de uno a muchos) y M:N (de muchos a muchos), dependiendo de la organización de los datos se optará por una u otra relación entre las tablas existentes.
- **Bases de datos no relacionales:** Ofrece un almacenamiento no estructurado y semi-estructurado de los datos. Estas son más usadas para páginas web por la cantidad de datos que pueden ser almacenados.

Para la gestión de la base de datos existen diferentes sistemas gestores de base de datos (DBMS, por sus siglas en inglés) que son conjuntos coordinados de programas, procedimientos y lenguajes que permiten a los usuarios realizar consultas a los datos almacenados. Por ejemplo Microsoft Access y MySQL, para bases de datos relacionales y MongoDB para no relacionales,

entre otros.

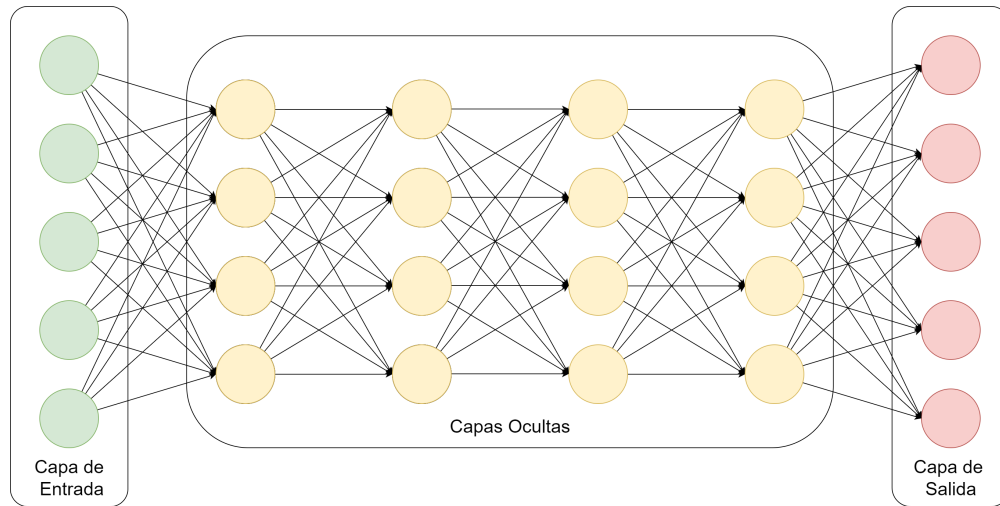
1.3. Machine Learning

El aprendizaje automático o aprendizaje de máquinas es una rama de la inteligencia artificial cuyo objetivo principal es permitir el desarrollo de algoritmos para la identificación de patrones en datos masivos y elaborar predicciones respecto al procedimiento a realizar. Esta técnica ha ganado relevancia debido al aumento de la capacidad de los procesadores, permitiendo realizar tareas específicas con cantidades amplias de datos, sin necesidad de ser programados propiamente para ello.

Dentro de los diferentes tipos de aprendizaje de máquina conocidos actualmente se encuentran: el aprendizaje reforzado, el no supervisado y el supervisado. Tomando este último para el desarrollo del proyecto, ya que consiste en el entrenamiento de un modelo de inteligencia artificial del cual se conoce la respectiva etiqueta de los dispositivos. Dentro del aprendizaje supervisado se encuentran varios tipos de metodologías de clasificación y regresión. Dada la salida requerida, es decir, la identificación de cada aparato, se establece el problema a resolver como uno de clasificación, para el cual se toman los siguientes métodos para su futura evaluación de desempeño.

1.3.1. Deep Neural Networks. Las redes neuronales profundas (DNN, por sus siglas en inglés) son algoritmos basados en el cerebro humano. Cada nodo o “neurona” en la red neuronal solo puede realizar cálculos simples, pero los DNN funcionan conectando los nodos para formar una serie de capas ocultas, donde cada capa realiza un cálculo basado en la salida de la capa anterior, es decir, el número de capas ocultas puede ser muy grande. De esta manera, pueden realizar

Figura 1. Arquitectura de una DNN.



tareas que son mucho más complejas de lo que un solo nodo podría lograr (Toh, Dondelinger, y Wang, 2019). Un ejemplo gráfico de esta arquitectura se presenta en la Figura 1, en el cual se observa la interconexión de las capas ocultas.

1.3.2. Support Vector Machine. Las máquinas de soporte vectorial (SVM, por sus siglas en inglés) son un conjunto de algoritmos cuyo proceso general es el siguiente: los datos comienzan en una dimensión relativamente baja, se procede a moverlos a una dimensión mayor que permita al clasificador de soporte vectorial separar los datos en los grupos requeridos a través de un hiperplano cuyos parámetros son encontrados sistemáticamente a través de la función del kernel. Existen principalmente dos tipos de kernel para esta clasificación:

- **Kernel polinomial:** La expresión matemática que rige este procedimiento se presenta en la Ecuación 1, las variables a y b refieren a dos diferentes observaciones en el conjunto de datos, r determina el coeficiente del polinomio y d el grado del mismo, son estos dos últimos

parámetros los que se configuran debidos al barrido de datos para configurar el hiperplano que mejor divida los conjuntos solicitados.

$$(a * b + r)^d \quad (1)$$

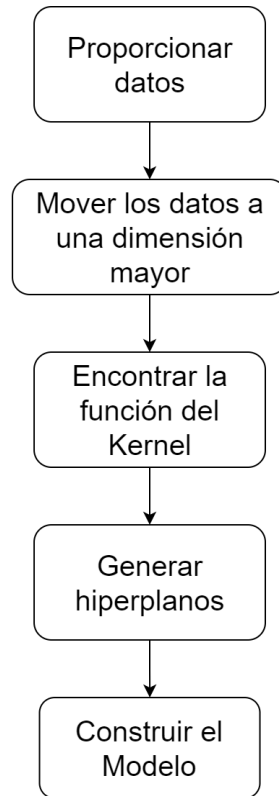
- Kernel radial o RBF:** La definición de este kernel se presenta en la Ecuación 2, las variables a y b refieren, al igual que en el anterior kernel, a dos diferentes observaciones en el conjunto de datos, en esta función la diferencia entre estos dos datos es duplicada, brindando la distancia entre ambas observaciones, la cual se multiplica con el parámetro γ , valor que escala la influencia de la distancia anteriormente calculada y el producto resultante es sumado al parámetro de regularización C el cual controla el margen del hiperplano.

$$e^{-\gamma(a-b)^2+C} \quad (2)$$

El proceso que lleva a cabo el algoritmo ha sido descrito mediante un diagrama de flujo, el cual se encuentra en la Figura 2.

1.3.3. Validación cruzada. La validación cruzada o cross-validation es una técnica utilizada para validar modelos generados en el aprendizaje automático, la cual consiste en calcular la media aritmética de las medidas de evaluación sobre diferentes particiones del conjunto de datos. Su propósito es garantizar que los resultados obtenidos sean independientes de la partición

Figura 2. Algoritmo SVM.

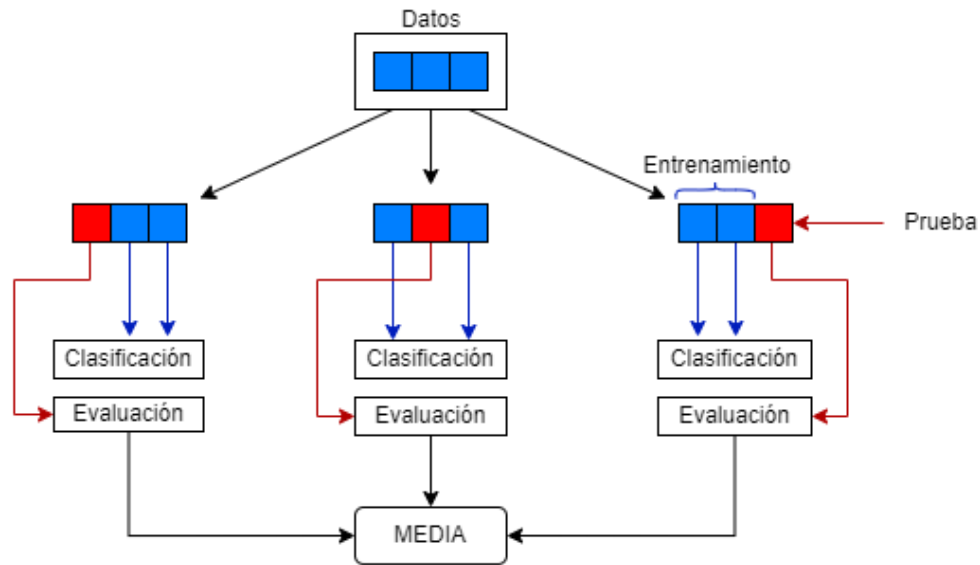


entre datos de entrenamiento y datos de validación. Se utiliza en entornos donde el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica.

Un ejemplo gráfico de esta técnica se presenta en la Figura 3, en el cual se observa el esquema de la validación cruzada con tres iteraciones y un solo clasificador.

1.3.4. Evaluación de modelos de clasificación. Después de entrenar un modelo es necesario cuantificar su rendimiento, por lo tanto, se utilizan determinadas métricas que permiten compararlo con otros modelos. A continuación se presentan las métricas utilizadas en este proyec-

Figura 3. Esquema de la validación cruzada.



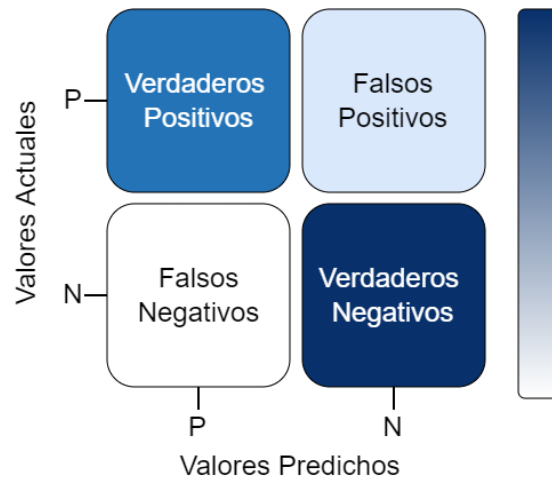
to:

- Matriz de confusión:** La matriz de confusión o matriz de error consiste en un resumen presentado en tablas donde se puede ver el número de predicciones correctas e incorrectas divididas en las diferentes categorías.

Se presenta en la Figura 4 un ejemplo en el que se tienen dos clases, la positiva y negativa, representados por las letras 'P' y 'N' respectivamente, el desempeño es medido de la siguiente manera:

- Verdadero positivo (VP): Indica que el modelo predijo correctamente una clase positiva.
- Verdadero negativo (VN): Indica que el modelo predijo correctamente una clase negativa.

Figura 4. Estructura de una Matriz de Confusión 2x2



- Falso positivo (FP): Indica que el modelo predijo erróneamente una clase positiva cuando realmente era negativa.
- Falso negativo (FN): Indica que el modelo predijo erróneamente una clase negativa cuando realmente era positiva.

Además de los términos claves que se usan para medir gráficamente el desempeño, se tiene una clasificación por color con una barra a la derecha del diagrama, que indica la cantidad de elementos clasificados correctamente en cada categoría.

- **Exactitud:** Se define como la relación entre la cantidad de predicciones que el modelo clasificó correctamente y el total de predicciones realizadas, tal como se observa en la Ecuación 3.

$$\text{Exactitud} = \frac{\text{Predicciones Correctas}}{\text{Total Predicciones}} = \frac{VP + VN}{VP + VN + FP + FN} \quad (3)$$

- **Precisión:** Se define como la dispersión del conjunto de valores acertados por clase, por ejemplo, el número de identificaciones positivas que fueron correctas relacionadas con el total de identificaciones, como se presenta en la Ecuación 4.

$$\text{Precisión} = \frac{VP}{VP + FP} \quad (4)$$

- **ROC-AUC:** Se define como el área bajo la curva (AUC por sus siglas en inglés) de la gráfica de características de funcionamiento del receptor (ROC por sus siglas en inglés).

Para graficar la curva ROC hay que tener en cuenta la tasa de verdaderos positivos (TVP) o sensibilidad, la especificidad y la tasa de falsos positivos (TFP), cuyas expresiones se presentan en las ecuaciones 5, 6 y 7.

$$\text{TVP} = \frac{VP}{VP + FN} \quad (5)$$

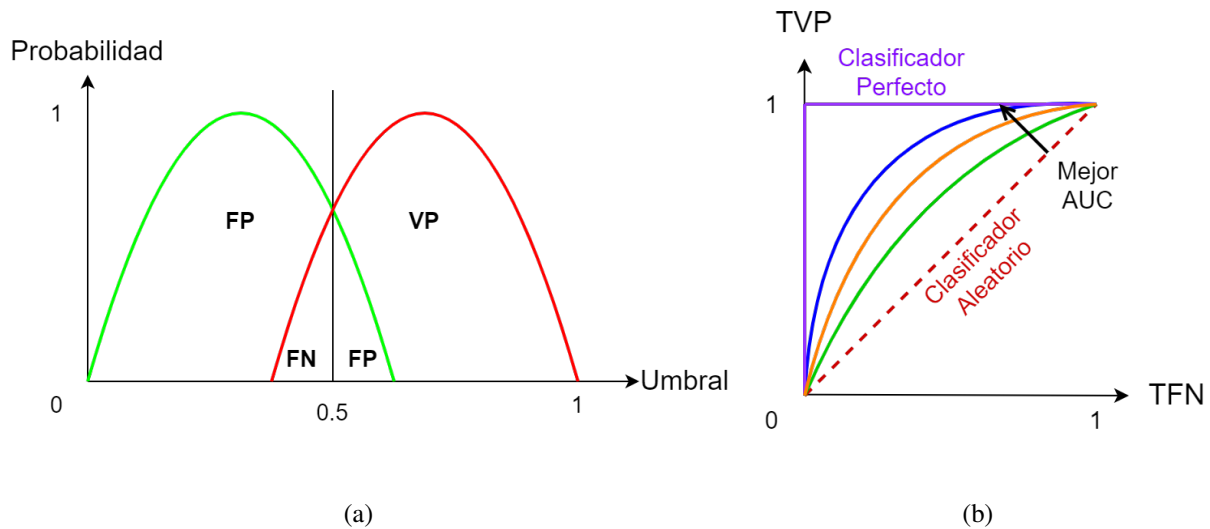
$$\text{Especificidad} = \frac{VN}{VN + FP} \quad (6)$$

$$\text{TFP} = 1 - \text{Especificidad} = \frac{FP}{VN + FP} \quad (7)$$

En la Figura 5 se puede observar cómo se representa la métrica, la cual toma las probabili-

dades presentadas en la Figura 5(a) para la distribución de la curva ROC de la Figura 5(b) en la cual define el área que determina el parámetro a utilizar.

Figura 5. Representación de la ROC-AUC: (a) Distribución normal de probabilidad para clasificaciones positivas y negativas, (b) Curva ROC

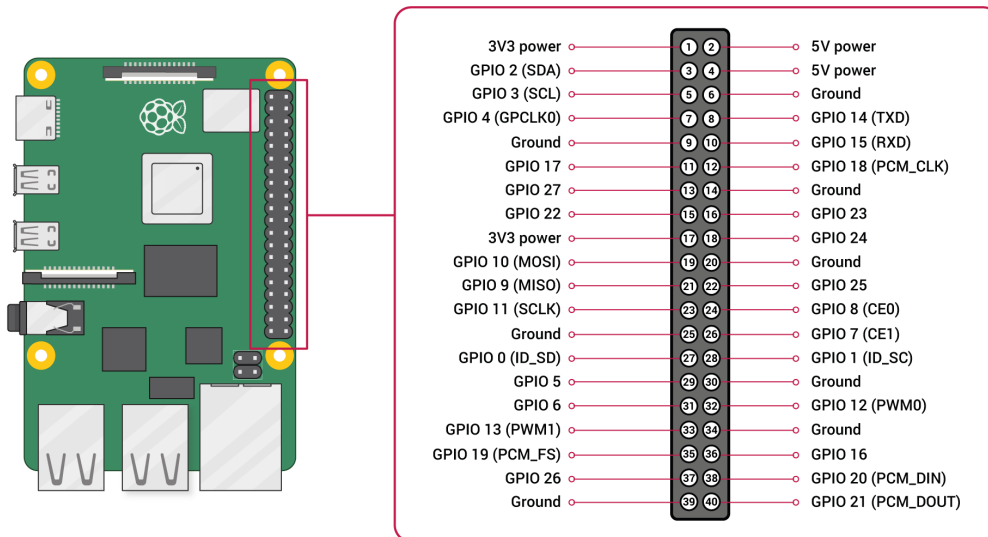


1.4. Raspberry Pi 3 B+

La Raspberry Pi es un computador muy versátil, de bajo costo, que puede conectarse a un monitor, permitiendo la exploración de recursos informáticos con la posibilidad de operar con una versión ligera del sistema operativo Linux denominada Raspbian.

Esta tarjeta ha sido elegida para la implementación de los algoritmos desarrollados ya que tiene grandes características técnicas para ejecutar fases de proyectos de inteligencia artificial e implementaciones en procesos de análisis o IoT. El modelo usado para este proyecto es la Raspberry Pi 3 B+ presentada en la Figura 6 la cual cuenta con 40 pines, entre los cuales hay 26 GPIO (puer-

Figura 6. Pines Raspberry Pi 3 B+



Fuente: <https://www.raspberrypi.com/documentation/computers/os.html>

tos de entrada o salida) que permiten la conexión de sensores, entre otros elementos útiles para el presente proyecto. Además de esto, al tener una memoria SDRAM de 1Gb da la posibilidad de almacenar los datos para su procesamiento, trabajo para el cual también cuenta con recursos suficientes de velocidad y conectividad para lograr un buen desempeño (*Raspberry Pi Documentation*, s.f.).

1.5. Angular CLI

Es un framework especializado en aplicaciones web que, a través de una interfaz de línea de comandos, permite el desarrollo ordenado del back-end y front-end, brindando herramientas de código abierto útiles para el uso de plantillas personalizadas para la creación de gráficos, tablas, menús de

navegación, entre otros (*Angular Documentation*, s.f.).

1.6. Google Firebase

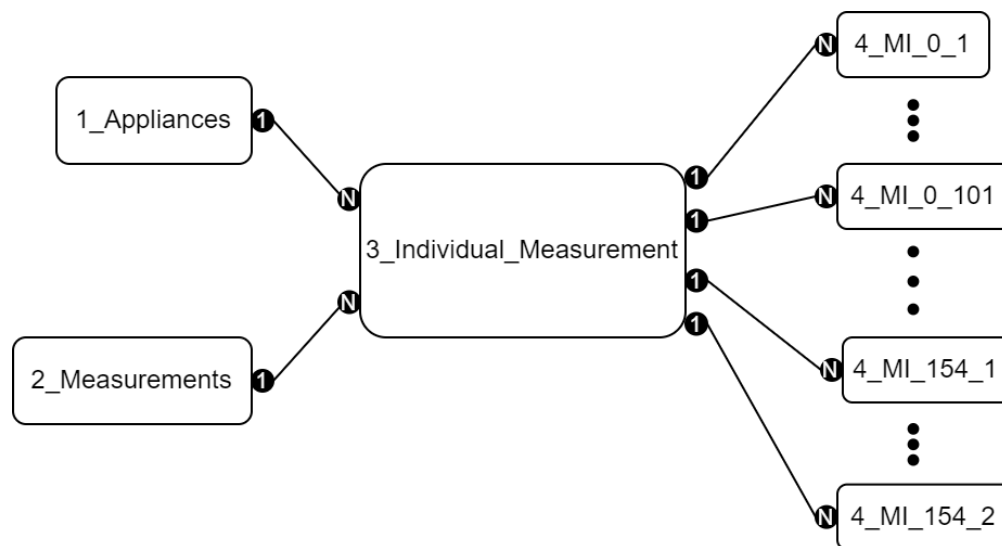
Es una plataforma en la nube creada para el desarrollo de aplicaciones web y móvil. Brinda diferentes servicios de analítica (brindando información estadística pertinente a la aplicación), autenticación (brindando servicios para la validación de los usuarios registrados), bases de datos en tiempo real, funciones en la nube (permitiendo la ejecución automática del back-end a peticiones HTTPS) y el alojamiento de las páginas web (*Guías de Firebase*, s.f.).

2. Base de datos

La base de datos se ha diseñado bajo una estructura relacional debido a la naturaleza de la información, la cual ha sido recolectada en el año 2016 por el grupo de investigación GISEL de la Universidad Industrial de Santander (Jiménez, 2018).

Tomando como base solamente las mediciones individuales de tensión y corriente de los electrodomésticos, se utiliza el gestor MySQL para construir el banco de datos. En este se diseña la arquitectura presentada en la Figura 7, la cual representa la información que se describe a continuación:

Figura 7. Arquitectura de la base de datos relacional



2.1. Electrodomésticos

La Figura 8 presenta una muestra de la codificación de todos los electrodomésticos usados para realizar las mediciones, teniendo en cuenta la potencia nominal de cada uno y su respectiva marca. La columna llamada “code” es la clave primaria de la tabla, ya que es la que identifica cada una de las filas con un código único.

Figura 8. Muestra de la tabla 1_Appliances

#	code	appliance	power	brand
1	0	Blender	450	Oster
2	1	Blender	600	Oster
3	2	Blender	600	Osterizer
4	3	Bulb	75	Sylvania
5	4	Cellphone Charger	36	Samsung
6	5	Cellphone Charger	36	Speed song
7	6	Cellphone Charger	20	Motorola
8	7	Cellphone Charger	24	Motorola
9	8	CFL	9	Interlight
10	9	CFL	20	Daiku
11	10	Desktop PC	250	Dell

2.2. Mediciones

La Figura 9 presenta una muestra de la codificación de todos los tipos de mediciones implementados, teniendo en cuenta si se realizó con cables largos y la impedancia asociada a estos, el estado en el que se encontraba el aparato dependiendo de los modos de operación del mismo y por último si la medición se realizó en la red (grid o G), con una fuente senoidal (sine o S) o con una onda plana (flat o F). Al igual que en la tabla anterior, la columna llamada “code” es la clave primaria.

Figura 9. Muestra de la tabla 2_Measurements

#	code	long_wires	state	grid_sine_flat
1	0	No	2	F
2	1	No	1	F
3	2	No	1	G
4	3	No	2	G
5	4	No	0	G
6	5	No	3	G
7	6	No	4	G
8	7	No	1	S
9	8	No	2	S
10	9	No	0	S
11	10	No	3	S

2.3. Mediciones individuales

La Figura 10 presenta una muestra de la codificación de cada medición realizada, teniendo en cuenta la fecha, el aparato utilizado, el tipo de medición y el número de mediciones realizadas. Se usan llaves foráneas, las cuales se encargan de hacer las relaciones pertinentes, en este caso, enlazar el código del aparato y medición a la nueva tabla, además cada elemento de esta, también tiene un identificador presentado en la columna “id”.

2.4. Tensión - Corriente

La Figura 11 representa un fragmento de una de las 154 tablas que almacenan los datos de tensión y corriente medidos en cada evento, incluye una variable “m” la cuál enlaza con la medición individual a la que corresponden esos datos.

Figura 10. Muestra de la tabla 3_Individual_Measurement

#	id	date	appliance	measurement	length
1	0	2016-02-26	9	2	101
2	1	2016-02-26	8	2	101
3	2	2016-02-26	16	2	101
4	3	2016-02-26	25	2	101
5	4	2016-02-29	9	7	101
6	5	2016-02-29	8	7	101
7	6	2016-02-29	12	2	101
8	7	2016-02-29	12	7	102
9	8	2016-02-29	12	7	102
10	9	2016-02-29	16	7	101
11	10	2016-02-29	17	2	101

Figura 11. Muestra de la tabla 4_MI_0_1

#	m	voltage	current
1	0	180.123	-0.002554
2	0	180.436	-0.002706
3	0	180.508	-0.002938
4	0	180.753	-0.002597
5	0	180.955	-0.002512
6	0	180.998	-0.002534
7	0	181.305	-0.002835
8	0	181.319	-0.003436
9	0	181.432	-0.003126
10	0	181.602	-0.002223
11	0	181.544	-0.002845

3. Identificación de electrodomésticos

La base funcional de este proyecto se encuentra en los algoritmos de desagregación de carga desarrollados para la Tesis Doctoral (Jiménez, 2018) y traducidos a Python en trabajos previos (Salgado y Ávila, 2021) en los cuales se realiza la extracción de 8 características principales entre las cuales se encuentran: el factor de cresta, desviación estándar, media, curtosis, asimetría, entropía, duración y punto en la onda de conmutación. Estos algoritmos son tomados como base para el desarrollo del modelo de clasificación con máquinas de soporte vectorial, el cuál va a ser comparado con el modelo de redes neuronales profundas existente, variando el conjunto de datos de entrenamiento y evaluación.

3.1. Algoritmos en ambiente de prueba

Para verificar los algoritmos se utilizó un ambiente de pruebas proporcionado por Google, conocido como Colab o Colaboratory. Esta herramienta permite programar y ejecutar scripts de Python en cualquier navegador de forma sencilla y presenta ventajas como no requerir ningún tipo de configuración, dar acceso gratuito a GPUs y compartir contenido fácilmente.

3.2. Datos de prueba

Para realizar las pruebas se utilizaron dos escenarios en los cuales se varía el conjunto de datos utilizado para entrenar y evaluar el respectivo modelo.

1. **PLAID:** Conjunto de datos público (Gao, Giri, Kara, y Bergés, 2014) el cuál contiene las muestras presentadas en la Tabla 1.

Tabla 1
Base de Datos PLAID

Aparato	Número de Muestras
Aire Acondicionado	208
Aspiradora	73
Bombilla Incandescente	148
Calentador	85
Lámpara Fluorescente Compacta (CFL)	220
Laptop	207
Lavadora	75
Microondas	229
Secador de Cabello	248
Ventilador	210
Total	1793

2. **GISEL:** Base de datos del grupo de investigación GISEL de la Universidad Industrial de Santander, cuyas mediciones fueron hechas para la Tesis Doctoral (Jiménez, 2018). Para esta base de datos es importante resaltar que no se logró incluir todas las muestras dada a la limitación de almacenamiento en la nube, sin embargo, se procuró generar una muestra representativa para tener una equivalencia con respecto a la predecesora. Las muestras respectivas por electrodoméstico son presentadas en la Tabla 2.

3.3. Ejecución en ambiente de prueba

Se eligieron tres electrodomésticos (ventilador, secador de cabello y portátil) con un número equivalente de datos para las pruebas, garantizando que los clasificadores sean entrenados equitativamente, ya que un electrodoméstico con más muestras que los otros puede provocar un mejor desempeño del modelo para ese caso particular. Cabe mencionar que la red neuronal profunda en-

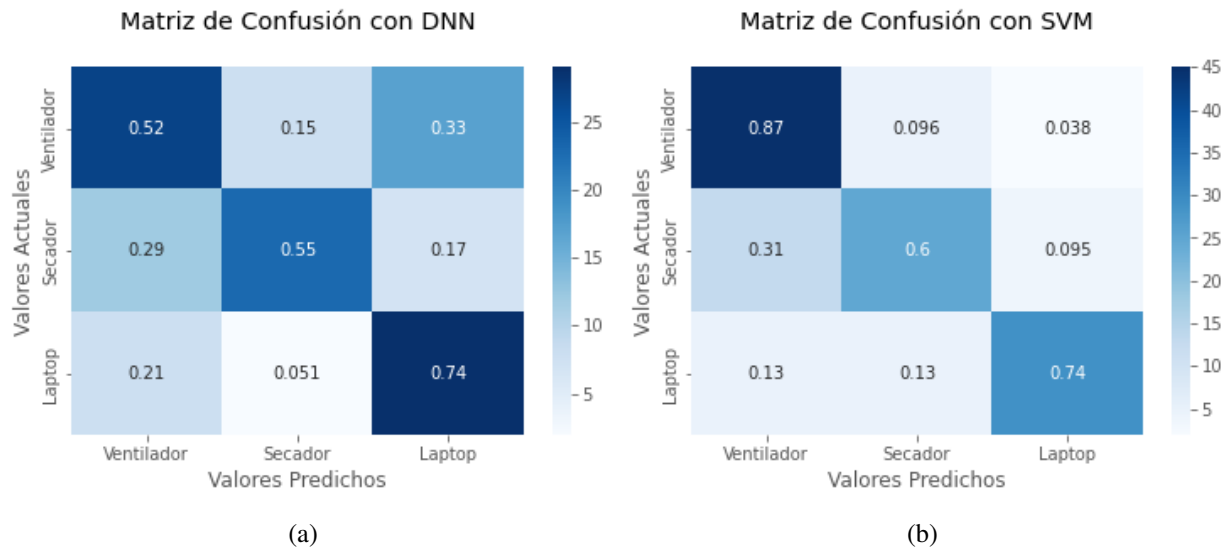
Tabla 2
Base de Datos GISEL

Aparato	Número de Muestras
Bombilla	231
Cargador de Celular	233
Cargador de Laptop	230
Computador de Escritorio	231
Lámpara Fluorescente Compacta (CFL)	261
Lámpara Halógena	231
Lámpara LED	220
Licuadaora	230
Nevera	213
Plancha	215
Sandwichera	248
Secador de Cabello	240
Televisor de Rayos Catódicos	207
Ventilador	220
Total	3210

trenada para este proyecto se creó en un trabajo previo con dos capas ocultas de 30 y 10 nodos respectivamente (Salgado y Ávila, 2021). Adicionalmente, el modelo para las maquinas de soporte vectorial usó un rango de parámetros para C y γ de $[0, 5 - 100]$ y $[1 - 0,0001]$ respectivamente, los cuales son configurados automáticamente por la red empleando el método de la validación cruzada.

La matriz de confusión de cada uno de los modelos de clasificación en el primer escenario se observa en la Figura 12, en estas se puede observar que el elemento mejor detectado haciendo uso del modelo con DNN es la laptop con un 74% de precisión y para el modelo con SVM es el ventilador con el 87%.

Figura 12. Matrices de confusión para el primer escenario: (a) Modelo con DNN, (b) Modelo con SVM



Las gráficas ROC para este primer escenario se presentan en la Figura 13, en ellas se observa el desempeño de cada uno de los electrodomésticos respecto a los otros. De manera general, el AUC general para el modelo con DNN tiene un valor de 0,7 y para el SVM de 0,86.

La Tabla 3 presenta las métricas utilizadas para evaluar los clasificadores, se puede observar que el modelo basado en DNN tuvo un resultado de 0,59 en exactitud, mientras que el modelo basado en SVM tuvo un resultado de 0,74, es decir, que este último presenta una mejora del 15% al momento de clasificar los electrodomésticos.

Figura 13. Curvas ROC para el primer escenario: (a) Modelo con DNN, (b) Modelo con SVM

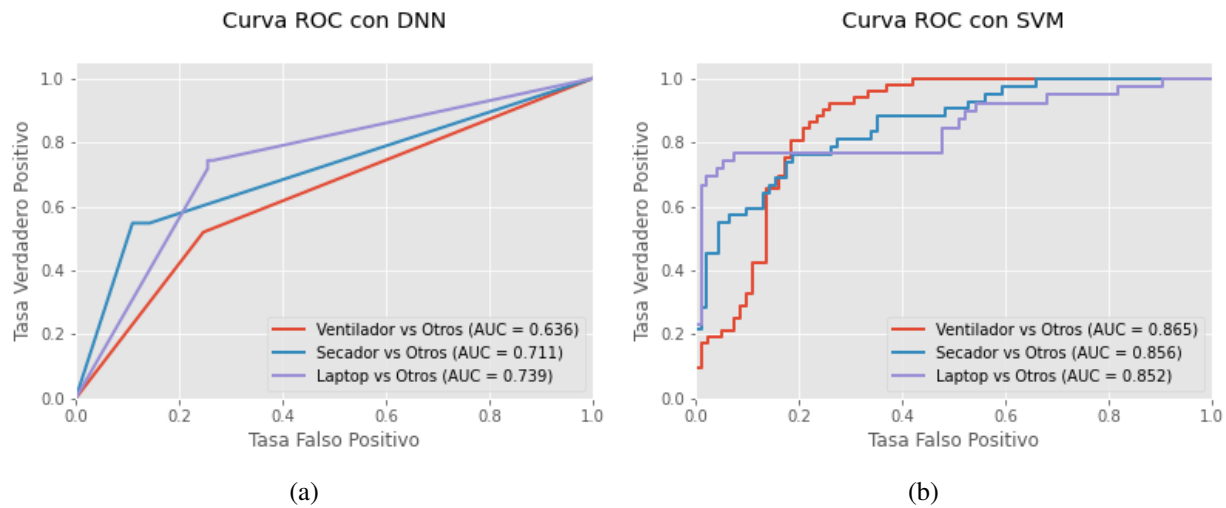


Tabla 3

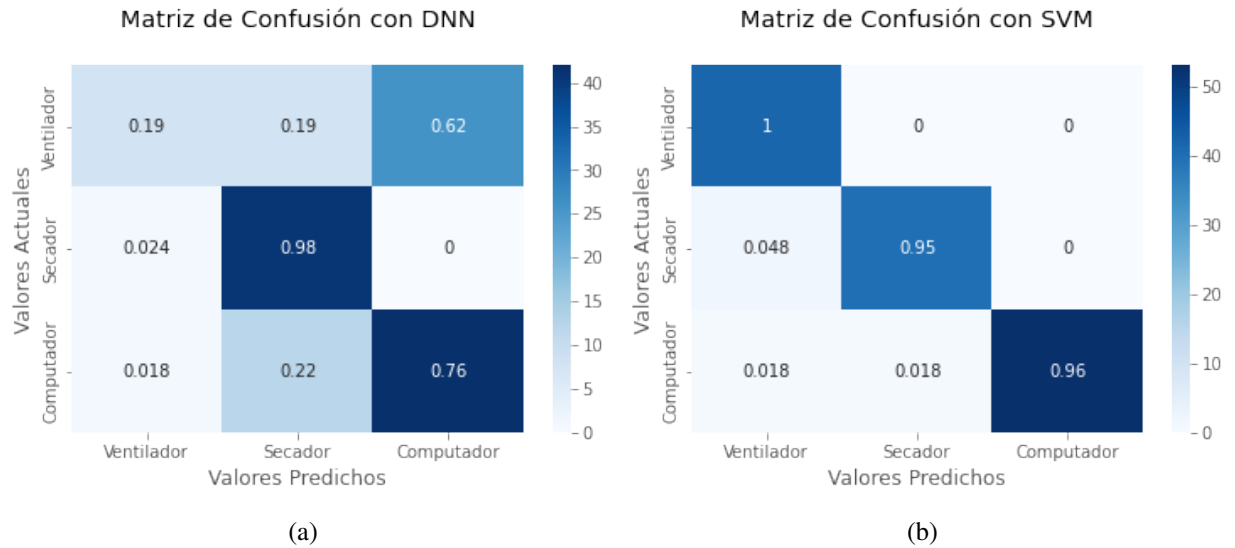
Resultados de los modelos de clasificación en el primer escenario

Modelo	Métrica	Electrodoméstico		
		Ventilador	Secador de Cabello	Portátil
DNN	Precisión	0,52	0,55	0,74
	AUC	0,64	0,71	0,74
	Exactitud	0,59		
SVM	Precisión	0,85	0,6	0,74
	AUC	0,87	0,86	0,85
	Exactitud	0,74		

El comportamiento de las matrices de confusión de los modelos de clasificación en el segundo escenario se observa en la Figura 14, en donde se nota como ambos modelos aumentan su desempeño y sus clasificaciones correctas toman más peso en el proceso. Este cambio es debido a que las señales utilizadas están más homogeneizadas respecto a su cantidad de elementos, lo cuál hace que a pesar de que las señales sean reales y tengan ruido, los clasificadores logren reconocer

cada uno de los elementos.

Figura 14. Matrices de confusión para el segundo escenario: (a) Modelo con DNN, (b) Modelo con SVM



Las curvas ROC para el segundo escenario se presentan en la Figura 15, y al igual que en el primer escenario, el desempeño de cada uno de los electrodomésticos se define respecto a los otros. El AUC general para el modelo con DNN tiene un valor de 0,73 y para el SVM es de 0,99.

La Tabla 4 presenta las métricas utilizadas para evaluar los clasificadores. En esta prueba, el modelo basado en SVM continua obteniendo una mejora al momento de identificar los electrodomésticos con respecto al modelo basado en DNN.

Figura 15. Curvas ROC para el segundo escenario: (a) Modelo con DNN, (b) Modelo con SVM

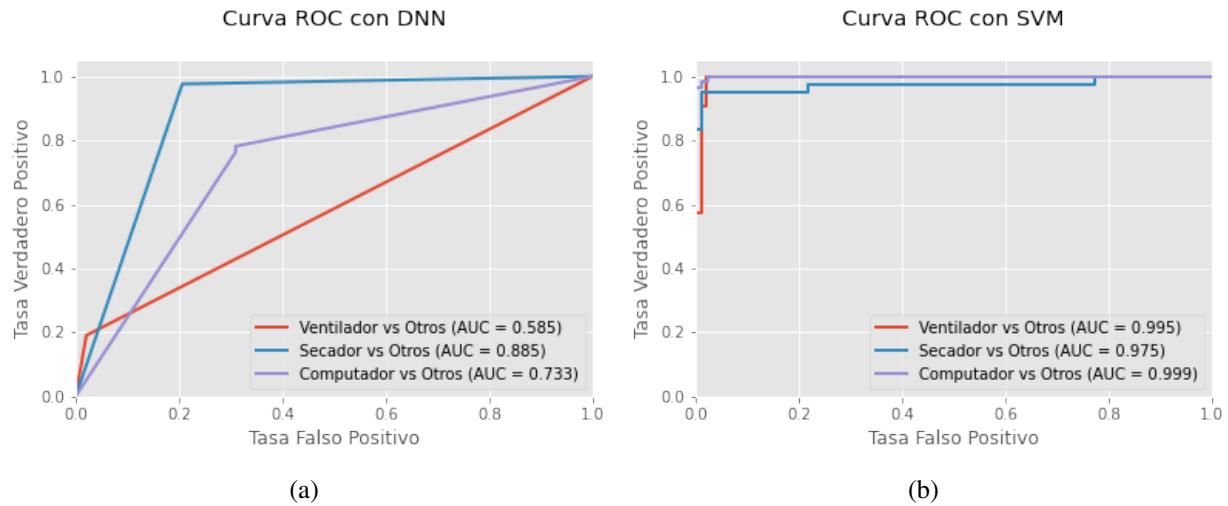


Tabla 4

Resultados de los modelos de clasificación en el segundo escenario

Modelo	Métrica	Electrodomésticos		
		Ventilador	Secador de Cabello	Laptop
DNN	Precisión	0,19	0,98	0,76
	AUC	0,59	0,89	0,73
	Exactitud		0,66	
SVM	Precisión	1	0,95	0,96
	AUC	1	0,98	1
	Exactitud		0,97	

3.4. Ejecución en el sistema destino

Una vez finalizadas las pruebas, el siguiente paso consistió en compilar toda la información en la tarjeta Raspberry Pi 3 B+. Para esto, se instaló el sistema operativo Raspbian Pi Os (32-bit), el cual cuenta con los programas básicos de un sistema operativo basado en Linux.

Para la migración de los códigos a la tarjeta primero se instalaron las bibliotecas utilizadas, las cuales garantizan que el código sea leído y compilado correctamente. Después, la detección de eventos fue realizada, iniciando con la generación de características de las señales y prosiguiendo con el modelo de clasificación basado en SVM, verificando su funcionamiento con el ambiente de pruebas. Primeramente, se utilizó el modelo entrenado solo con 3 electrodomésticos (ventilador, secador y computador) brindando resultados satisfactorios de predicción del elemento.

La aplicación desarrollada incluye un fraccionamiento del proceso a desarrollar, el cuál comienza con la carga de datos, refiriendo a lo que podría ser la lectura directamente del sensor o en este caso, el archivo con las firmas de carga del electrodoméstico que se quiere identificar. Seguido de esto, se extraen los rangos de la señal para identificar el estado estable de la misma, para luego realizar el cálculo de la potencia y la extracción de características. Después la predicción es realizada con el vector de características obtenido y por último, se envía la señal de activación para que en la página web se actualice la conexión de este elemento. En las Figuras 16, 17 y 18 se visualizan los tiempos de ejecución de cada sección, entre los parámetros importantes para el seguimiento. El algoritmo utilizado para esta aplicación se presenta en el Apéndice 1

Figura 16. Ejecución del algoritmo en la Raspberry para el ventilador

```
monitore3t@raspberrypi: ~
Archivo Editar Pestañas Ayuda

NILM TEST
APPLICATION

-----

* PASO 1: Carga de Datos
Tiempo de carga de los datos 3.6374[s]

-----

* PASO 2: Extracción de Rangos
Tiempo de extracción de rangos 0.3288[s]

-----

* PASO 3: Calculo de Potencia
La potencia consumida por el dispositivo es 0.19[W]
Tiempo de cálculo de potencia 0.5315[s]

-----

* PASO 4: Extracción de Características
Tiempo de extracción de características 199.5548[s]

-----

* PASO 5: Creacion de la Matriz de Características
Dimensiones del Array 1: (1, 42)
Dimensiones del Array 2: (1, 280)
Dimensiones del Array 3: (1, 322)
Dimensiones del Array 4: (1, 63)
Dimensiones del Array 5: (1, 21)
Dimensiones del Array 6: (1, 84)
Dimensiones del Array 7: (1, 406)
Dimensiones del Array 8: (1, 42)
Dimensiones del Array 9: (1, 448)
Tiempo de creación de la matriz 1.3418[s]

-----

* PASO 6: Importe del Modelo
SVC(C=100, gamma=0.001, probability=True)
Tiempo de importacion del modelo 0.6840[s]

-----

* PASO 7: Predicción
La predicción con SVM es: 0
El dispositivo detectado es Ventilador
Tiempo de predicción 0.7478[s]

-----

* PASO 8: Conexion a la Pagina Web
Tiempo de subida a la BD 3.9264[s]

-----
```

Figura 17. Ejecución del algoritmo en la Raspberry para el secador de cabello

```
monitore3t@raspberrypi: ~
Archivo Editar Pestañas Ayuda

NILM TEST
APPLICATION

-----

* PASO 1: Carga de Datos
Tiempo de carga de los datos 3.6026[s]

-----

* PASO 2: Extracción de Rangos
Tiempo de extracción de rangos 0.3056[s]

-----

* PASO 3: Calculo de Potencia
La potencia consumida por el dispositivo es 0.01[W]
Tiempo de cálculo de potencia 1.0641[s]

-----

* PASO 4: Extracción de Características
Tiempo de extracción de características 198.2018[s]

-----

* PASO 5: Creacion de la Matriz de Características
Dimensiones del Array 1: (1, 42)
Dimensiones del Array 2: (1, 280)
Dimensiones del Array 3: (1, 322)
Dimensiones del Array 4: (1, 63)
Dimensiones del Array 5: (1, 21)
Dimensiones del Array 6: (1, 84)
Dimensiones del Array 7: (1, 406)
Dimensiones del Array 8: (1, 42)
Dimensiones del Array 9: (1, 448)
Tiempo de creación de la matriz 1.5197[s]

-----

* PASO 6: Importe del Modelo
SVC(C=100, gamma=0.001, probability=True)
Tiempo de importacion del modelo 0.6631[s]

-----

* PASO 7: Predicción
La predicción con SVM es: 1
El dispositivo detectado es Secador
Tiempo de predicción 0.7237[s]

-----

* PASO 8: Conexion a la Pagina Web
Tiempo de subida a la BD 5.6644[s]

-----
```

Figura 18. Ejecución del algoritmo en la Raspberry para el computador

```
monitore3t@raspberrypi: ~
Archivo Editar Pestañas Ayuda

NILM TEST
APPLICATION

-----

* PASO 1: Carga de Datos
Tiempo de carga de los datos 3.6395[s]

-----

* PASO 2: Extracción de Rangos
Tiempo de extracción de rangos 0.3038[s]

-----

* PASO 3: Calculo de Potencia
La potencia consumida por el dispositivo es 0.11[W]
Tiempo de cálculo de potencia 0.5332[s]

-----

* PASO 4: Extracción de Características
Tiempo de extracción de características 207.9579[s]

-----

* PASO 5: Creacion de la Matriz de Características
Dimensiones del Array 1: (1, 42)
Dimensiones del Array 2: (1, 280)
Dimensiones del Array 3: (1, 322)
Dimensiones del Array 4: (1, 63)
Dimensiones del Array 5: (1, 21)
Dimensiones del Array 6: (1, 84)
Dimensiones del Array 7: (1, 406)
Dimensiones del Array 8: (1, 42)
Dimensiones del Array 9: (1, 448)
Tiempo de creación de la matriz 1.2501[s]

-----

* PASO 6: Importe del Modelo
SVC(C=100, gamma=0.001, probability=True)
Tiempo de importacion del modelo 0.6775[s]

-----

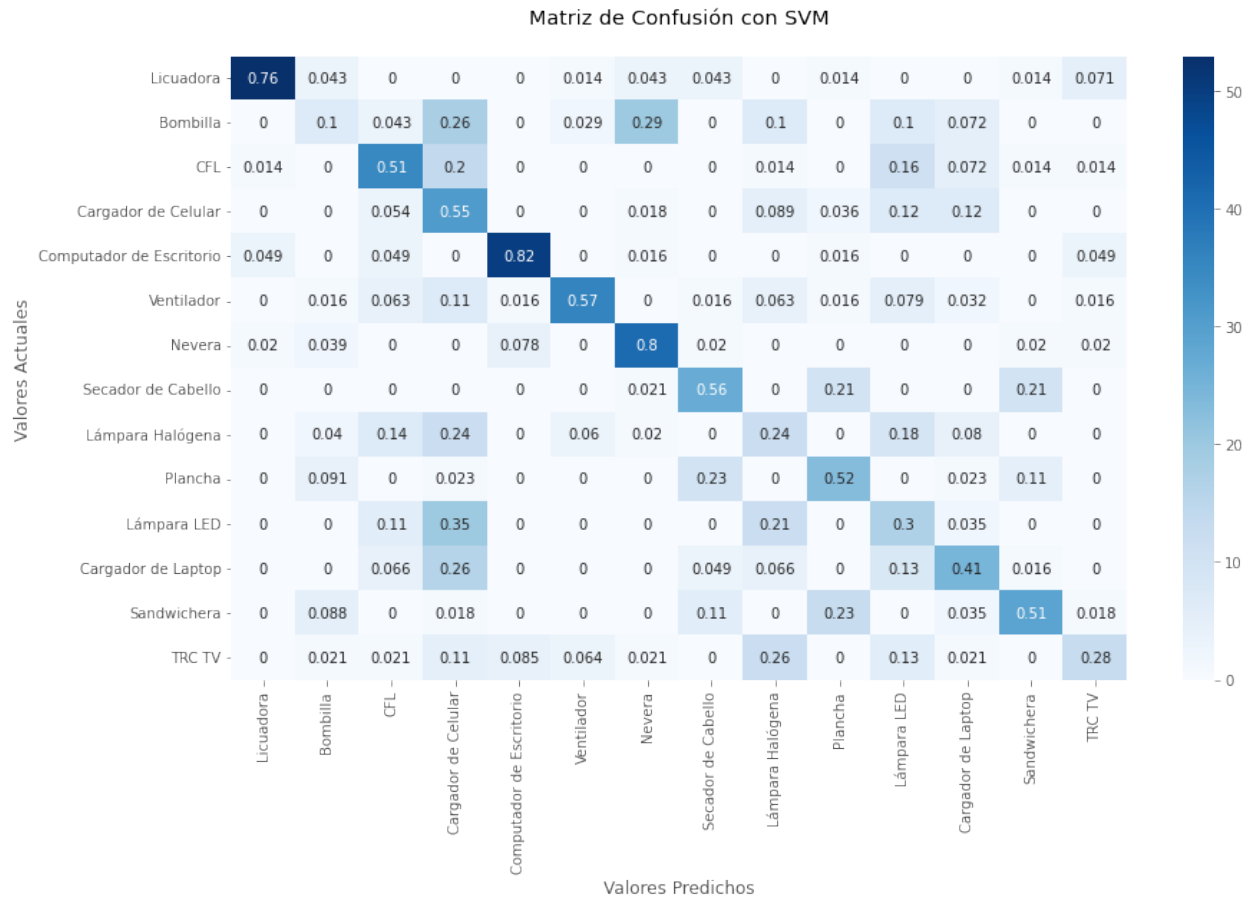
* PASO 7: Predicción
La predicción con SVM es: 2
El dispositivo detectado es Computador
Tiempo de predicción 0.7041[s]

-----

* PASO 8: Conexion a la Pagina Web
Tiempo de subida a la BD 4.0032[s]

-----
```

Figura 19. Matriz de confusión del modelo SVM en la Raspberry

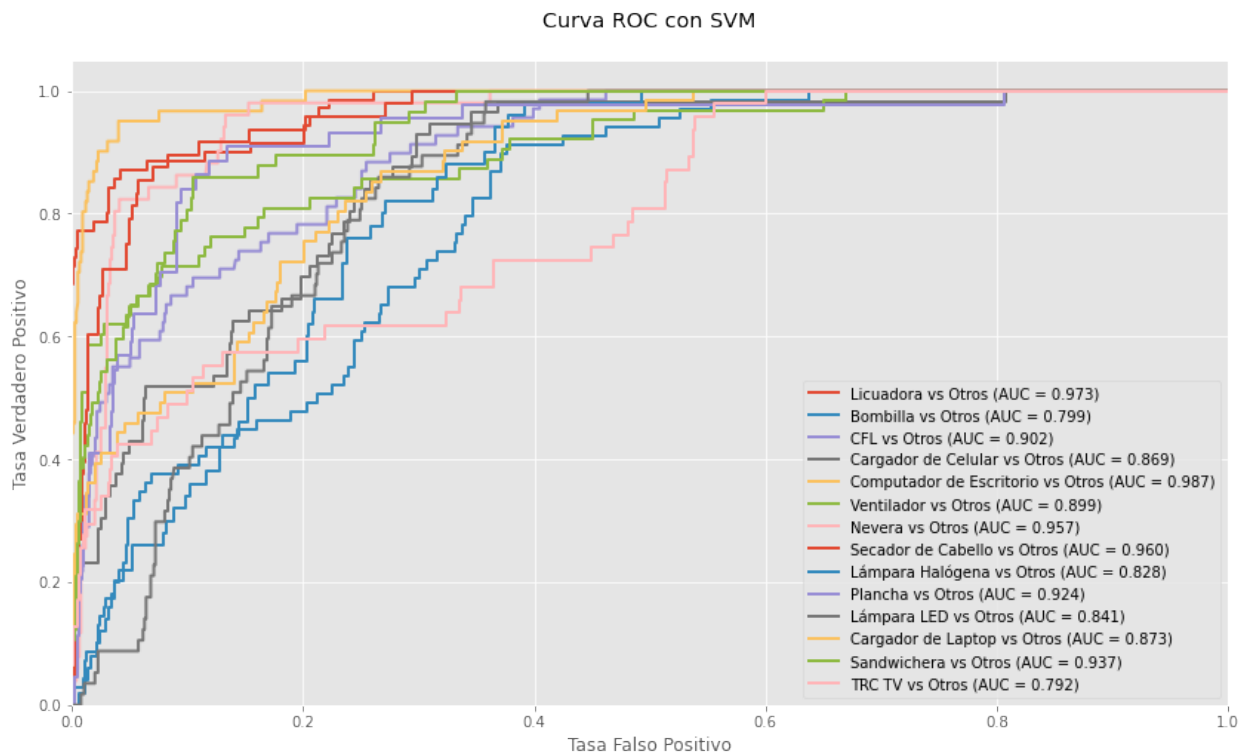


Seguido de la prueba con los 3 electrodomésticos anteriormente expuestos, y con la extracción de la matriz de características de toda la muestra de la base de datos de GISEL, se procede a evaluar el modelo para todos los elementos presentes. En la Figura 19 se visualiza el comportamiento de la matriz de confusión, donde se observan resultados satisfactorios, ya que se logra tener una diagonal marcada y precisiones hasta del 82%.

Por otra parte, las curvas ROC para el modelo se presentan en la Figura 20, para esta métrica

se observa un desempeño aceptable ya que ningún área se encuentra por debajo de 0,79 y de manera general, el AUC del modelo es del 90%. Es de destacar que el elemento con mayor AUC es el computador de escritorio con 0,987 y el menor es el televisor de rayos catódicos (TRC TV) con 0,792.

Figura 20. Curvas ROC del modelo SVM en la Raspberry



En la Tabla 5 se presentan los resultados de las métricas utilizadas para evaluar el modelo, en esta clasificación el electrodoméstico que mejor es detectado es el computador de escritorio con un 82% de precisión y un área bajo la curva de 0,987, seguido de la nevera que tiene 0,957 de AUC y una precisión del 80%. Por otro lado, el elemento más difícil de clasificar es la bombilla con un 10% de precisión. La exactitud general del modelo obtuvo un resultado del 50%, el cuál es

satisfactorio con solo una muestra de la base de datos.

Tabla 5

Resultados del modelo de clasificación SVM en la Raspberry

Electrodoméstico	Métrica		
	Precisión	AUC	Exactitud
Licuadaora	0,76	0,97	
Bombilla	0,10	0,79	
CFL	0,51	0,9	
Cargador de Celular	0,55	0,86	
Computador de Escritorio	0,82	0,98	
Ventilador	0,57	0,89	
Nevera	0,80	0,95	0,5
Secador de Cabello	0,56	0,96	
Lámpara Halógena	0,24	0,82	
Plancha	0,52	0,92	
Lámpara LED	0,30	0,84	
Cargador de Laptop	0,41	0,87	
Sandwichera	0,51	0,93	
TRC TV	0,28	0,79	

4. Aplicativo Web

Para poder visualizar los datos obtenidos por la Raspberry Pi se actualizó la página creada en el trabajo previo (Salgado y Ávila, 2021), en la cual se utilizaron las herramientas de Firebase por medio de una cuenta de Google.

4.1. Base de datos

Primero, se verificó que las tablas de la base de datos enviaran correctamente la información a la página web, esto se hizo por medio del lenguaje de programación Python. En la Tabla 6 se ilustra la estructura de las tablas de la base de datos.

Tabla 6

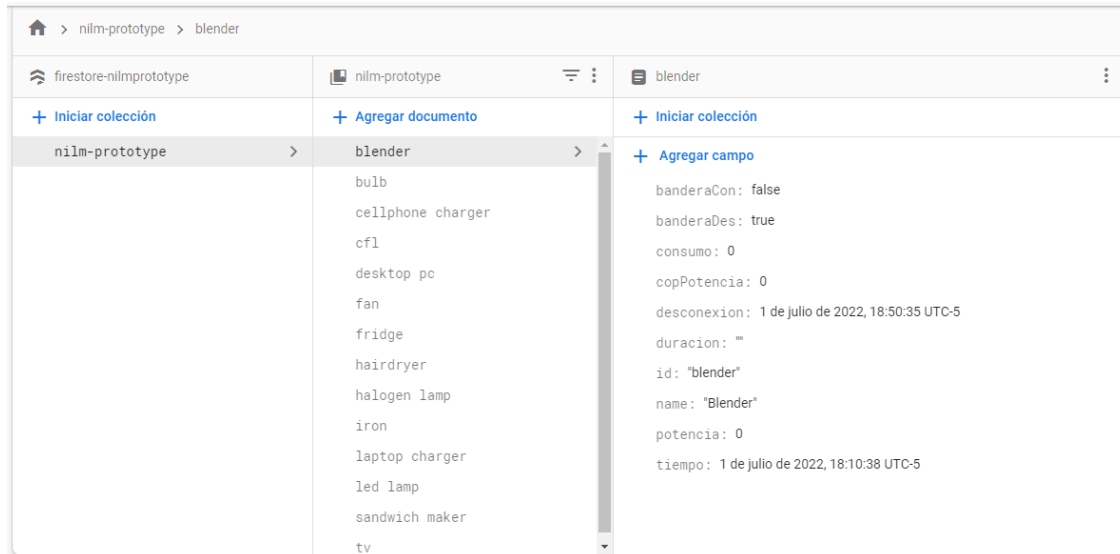
Estructura de la base de datos del aplicativo web

NILM-PROTOTYPE
banderaCon: booleano
banderaDes: booleano
consumo: number
copPotencia: number
desconexion: date
duracion: string
id: string
name: string
potencia: number
tiempo: date

Una vez finalizada la verificación, se utilizó Cloud Firestore para actualizar la base de datos no relacional en forma de colecciones. A esta se le crea un objeto asociado a cada electrodoméstico

como se presenta en la Figura 21.

Figura 21. Estructura de las colecciones.



4.2. Cálculo de la potencia

Un objetivo primordial de la página web es darle a los usuarios la información más relevante en diagramas de fácil entendimiento, para ello se calcula la potencia que consume cada electrodoméstico conectado, teniendo en cuenta que el tiempo de conexión también es obtenido por la aplicación, se genera una experiencia significativa para el usuario, habilitando información que lo lleve a tener un mejor control del consumo que realiza.

Para realizar el cálculo de la potencia activa se sigue la Ecuación 8 por lo cuál el primer paso para hallar su valor es tener definido el periodo de la señal. El periodo fundamental es extraído de la señal de tensión utilizando la Transformada Discreta de Fourier, y dado a que en Colombia

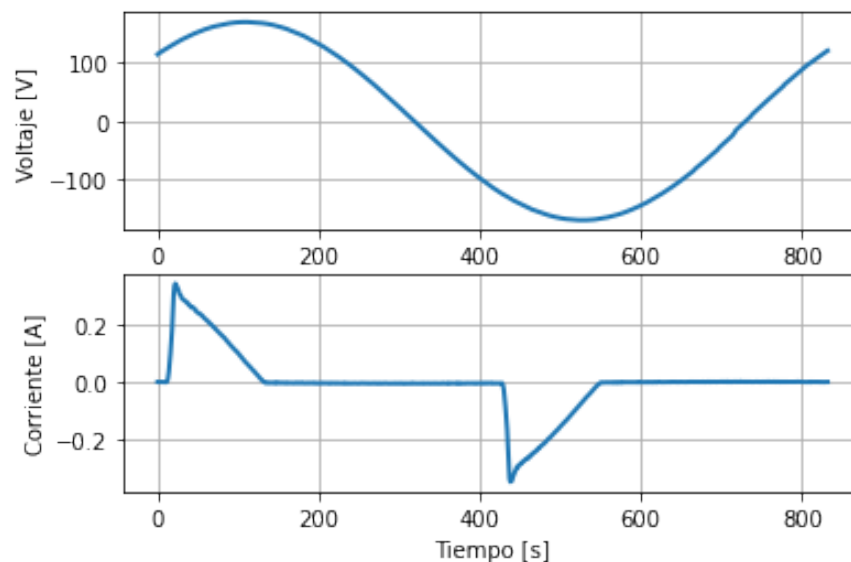
la frecuencia de la red está definida en 60[Hz] este valor encontrado teóricamente es verificado.

$$P = \frac{1}{N} \sum_{n=1}^N v[n]i[n] \quad (8)$$

Teniendo claro la extensión de la ventana que será operada, la cuál está dada por el periodo se procede a calcular el promedio de la tensión y corriente para poder hallar la potencia requerida.

Una visualización de la ventana mencionada para realizar estas operaciones se presenta en la Figura 22.

Figura 22. Tensión y corriente de un dispositivo CFL (id = 55) acotados en un periodo.



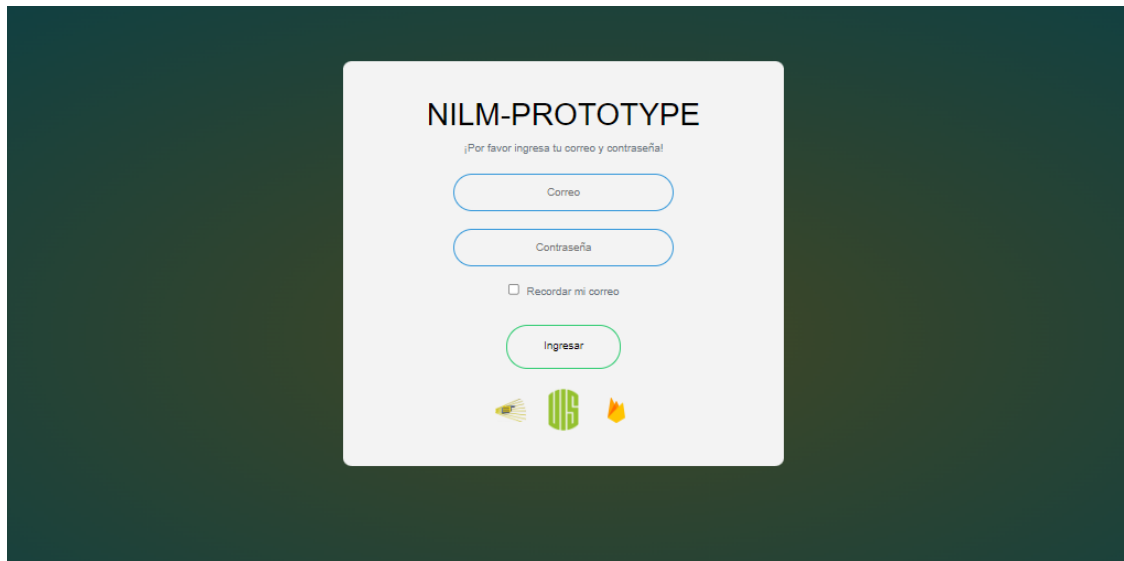
4.3. Visualización

La pagina web consta de una pantalla de login como se muestra en la Figura 23, en la cual solo se podrá ingresar con las credenciales correctas, de lo contrario aparecen alertas de notificaciones.

Además, tiene acceso directo a las páginas de la Universidad Industrial de Santander, Escuela de

Ingeniería Eléctrica, Electrónica y de Telecomunicaciones y de Firebase a través de sus respectivos logos.

Figura 23. Pantalla de login



Una vez se ha ingresado, se tienen dos pantallas para visualizar la información proporcionada por la tarjeta: Electrodomésticos y Potencia Total.

La sección Electrodomésticos contiene una gráfica de pastel indicando el consumo de cada uno de los electrodomésticos conectados como se observa en la Figura 24, así como una tabla que indica el estado, la potencia activa en tiempo real y la hora de conexión. Una vez todos los dispositivos se desconecten la gráfica de pastel desaparece como se observa en la Figura 25 y en la tabla se agrega la hora de desconexión, la duración y el consumo.

Figura 24. Sección Electrodomésticos con cuatro dispositivos conectados.

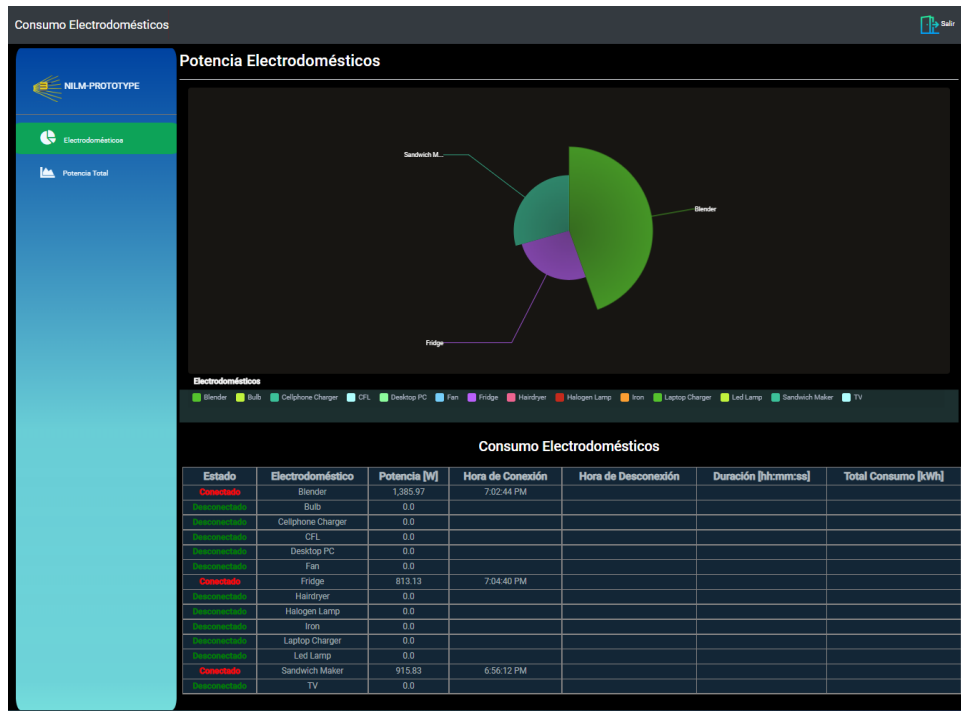
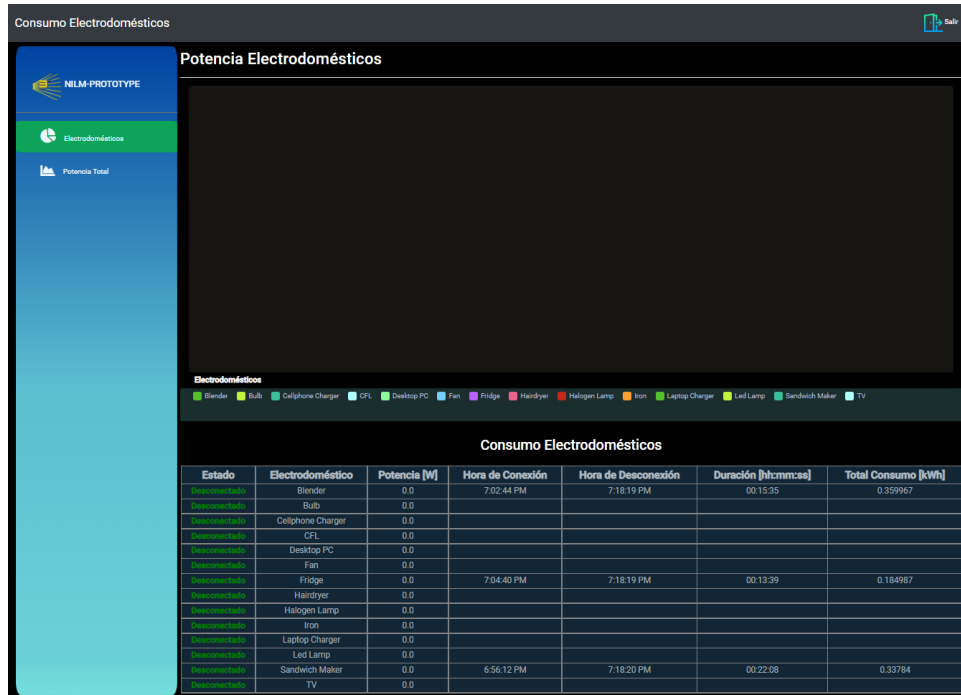
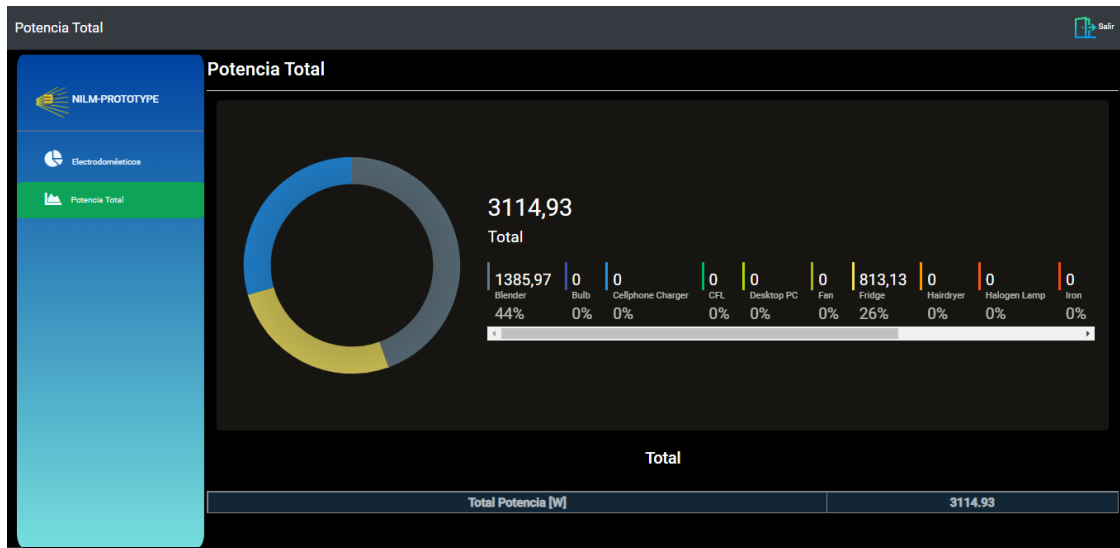


Figura 25. Sección Electrodomésticos con cuatro dispositivos desconectados.



La sección Potencia Total, como se ilustra en la Figura 26, contiene un gráfico circular con el porcentaje de consumo de cada electrodoméstico encendido. Además, indica la potencia activa total.

Figura 26. Sección Potencia Total.



A continuación, se proponen las siguientes recomendaciones para trabajos posteriores a la entrega de este proyecto.

5. Recomendaciones

- Para asegurar una base de datos funcional se propone la gestión de un servidor alojado en la nube para poder realizar consultas a la base de datos desde internet y poder enlazar esta a diferentes aplicativos. Además, se recomienda realizar el diseño relacional de la base de datos con las mediciones de electrodomésticos grupales.
- Con el propósito de mejorar los resultados de clasificación de todos los elementos se plantea un aumento en el número de muestras por dispositivo, incluyendo la base de datos de mediciones individuales completa. Es importante mencionar, que a medida que se aumente la cantidad de muestras, el tiempo que dura la extracción de características de las señales para organizar la matriz que se usa para el entrenamiento será mayor (actualmente con las muestras presentadas tiene una duración aproximada de 33 horas).
- Para el aplicativo web se propone incluir componentes más directos e importantes para el usuario, como un historial del consumo con el fin de realizar comparaciones y una estimación del costo mensual correspondiente al valor unitario de cada kilowatt, generando así la concientización deseada.

6. Conclusiones

En este documento se exploraron varias funcionalidades para las cuales el desarrollo de la base de datos fue de vital importancia. El hecho de alojar ordenadamente la información que se disponía en cada archivo, tuvo grandes implicaciones en el manejo, consulta y procesamiento de estos. Sin embargo, el espacio requerido para el almacenamiento fue una deficiencia ya que no se contaba con la capacidad suficiente para incluir toda la información recolectada.

Por otro lado, se analizaron distintos escenarios en la aplicación de redes neuronales profundas y máquinas de soporte vectorial para la desagregación de carga. Las pruebas brindan más resultados positivos para la implementación del modelo SVM en comparación al DNN ya que:

1. Presenta una mejora en la exactitud del 15 % con la base de datos PLAID y 30 % con la base de datos de GISEL, haciendo que las predicciones sean mas acertadas.
2. Muestra un desempeño mayor en un 15 % con la base de datos PLAID y 26 % con la base de datos de GISEL, en el área bajo la curva general del clasificador.
3. Tiene una alta tolerancia al ruido de las señales medidas ya que se obtuvieron mejores resultados con la base de datos de GISEL comparada con PLAID.
4. Es independiente de la selección del conjunto de prueba y evaluación gracias a la implementación de la validación cruzada.

Cabe destacar que para la cantidad de elementos tomados principalmente en este trabajo, la Raspberry Pi 3 B+ no cuenta con las condiciones necesarias de almacenamiento y RAM para desarrollar el procesamiento de datos y generar las características. Sin embargo, sigue siendo útil para calcular la potencia, y con el modelo ya entrenado, clasificar apropiadamente las señales haciendo uso de los algoritmos de detección de electrodomésticos desarrollados.

Además, la expansión de la página web permite al usuario tener un mayor control sobre los electrodomésticos comúnmente usados en el hogar, creando una conciencia al momento de utilizarlos sin necesidad.

Referencias Bibliográficas

- Angular documentation.* (s.f.). <https://angular.io/docs>.
- Darby, S. (2006). The effectiveness of feedback on energy consumption.
- Gao, J., Giri, S., Kara, E. C., y Bergés, M. (2014). Plaid: A public dataset of high-resolution electrical appliance measurements for load identification research: Demo abstract..
- Guías de firebase.* (s.f.). <https://firebase.google.com/docs/guides>.
- Jiménez, Y. (2018). *Automatic disaggregation of residential electrical consumption with non-intrusive methods.*
- Jiménez, Y., Cortes, J., Duarte, C., Petit, J., y Carrillo, G. (2014). Nonintrusive load monitoring for awareness of residential electricity consumption.
- Jiménez, Y., Duarte, C., Petit, J., Meyer, J., Schegner, P., y Carrillo, G. (2015). Steady state signatures in the time domain for nonintrusive appliance identification.
- Raspberry pi documentation.* (s.f.). <https://www.raspberrypi.com/documentation/computers/os>.
- Salgado, A., y Ávila, S. (2021). *Diseño e implementación de un monitor no intrusivo de potencia de electrodomésticos.*
- Toh, T. S., Dondelinger, F., y Wang, D. (2019, 08). Looking beyond the hype: Applied ai and machine learning in translational medicine. *EBioMedicine*, 47. doi: 10.1016/j.ebiom.2019.08.027
- World-Energy-Council. (2017). Escenarios energéticos mundiales.

Apéndices

Apéndice A. Código principal ejecutado en la Raspberry

```

import os
import numpy as np
import numpy.fft as fft
import time
import pandas as pd
import joblib
from sklearn.preprocessing import scale

import firebase_admin
from firebase_admin import credentials, firestore
import NILMfunctions
import art

os.system('cls')
print(art.title)
print(art.separator)

print('* PASO 1: Carga de Datos')
tic = time.perf_counter()
fs = 50000
FeaturesDataFrame = pd.read_csv('FeaturesDataFrame.csv')
InputData = np.genfromtxt('Data.csv', delimiter=',', names='voltage,current',
    dtype=(float,float))
Current = InputData['current']
Voltage = InputData['voltage']
toc = time.perf_counter()
tictoc = toc - tic
print(f'Tiempo de carga de los datos {tictoc:0.4f}[s]')
print(art.separator)

print('* PASO 2: Extracci n de Rangos')
tic = time.perf_counter()
TransientRange , SteadyRange = NILMfunctions.GetRanges(Current, 100, fs,
    0.9)
toc = time.perf_counter()
tictoc = toc - tic
print(f'Tiempo de extracci n de rangos {tictoc:0.4f}[s]')
print(art.separator)

print('* PASO 3: Calculo de Potencia')
tic = time.perf_counter()
N = len(Voltage)
temporal_window = N/fs
Ts = 60/(temporal_window/N)
a = np.abs(fft.rfft(Voltage, n=InputData.size))[1:]
periods = fft.rfftfreq(Voltage.size, d=1./Ts)[1:]
periods = np.divide(60, periods)
max_period = periods[np.argmax(a)]
v = []
c = []
n = -1
for i in np.linspace(0, temporal_window, Voltage.size):
    n += 1
    if (i >= temporal_window/2) and (i <= (temporal_window/2 + max_period)):
        v.append(Voltage[n])
        c.append(Current[n])
PowerCalc = round(abs(np.average(np.multiply(v,i))), 2)

```

```

print(f' La potencia consumida por el dispositivo es {PowerCalc}[W]')
toc = time.perf_counter()
tictoc = toc - tic
print(f'Tiempo de calculo de potencia {tictoc:0.4f}[s]')
print(art.separator)

print('* PASO 4: Extracci n de Caracteristicas')
tic = time.perf_counter()
FeatureV,Feature_time,Area,Feature1,Feature2,Feature3,Feature4,Feature7 =
    NILMfunctions.FinalIndivFeatureComputing5(Current,Voltage,
        TransientRange,SteadyRange,fs)
toc = time.perf_counter()
tictoc = toc - tic
print(f'Tiempo de extracci n de caracter sticas {tictoc:0.4f}[s]')
print(art.separator)

print('* PASO 5: Creacion de la Matriz de Caracteristicas')
tic = time.perf_counter()
DatasetInfo = []
DatasetInfo.append([Feature1, Feature2, Feature3, Feature4, Feature7])
col_names = ['FST1', 'FST2', 'FST3', 'FST4', 'FST7']
TestDF = pd.DataFrame(DatasetInfo, columns = col_names)
MasterArray = TestDF[['FST1', 'FST2', 'FST3', 'FST4', 'FST7']].to_dict('
    list')
Array1 = np.vstack(MasterArray['FST1'])
print(' Dimensiones del Array 1:',Array1.shape)
Array2 = np.vstack(MasterArray['FST2'])
print(' Dimensiones del Array 2:',Array2.shape)
Array3 = np.hstack([Array1, Array2])
print(' Dimensiones del Array 3:',Array3.shape)
Array4 = np.vstack(MasterArray['FST3'])
print(' Dimensiones del Array 4:',Array4.shape)
Array5 = np.vstack(MasterArray['FST4'])
print(' Dimensiones del Array 5:',Array5.shape)
Array6 = np.hstack([Array4,Array5])
print(' Dimensiones del Array 6:',Array6.shape)
Array7 = np.hstack([Array3,Array6])
print(' Dimensiones del Array 7:',Array7.shape)
Array8 = np.vstack(MasterArray['FST7'])
print(' Dimensiones del Array 8:',Array8.shape)
Array9 = np.hstack([Array7,Array8])
print(' Dimensiones del Array 9:',Array9.shape)
FeaturesArray = Array9
Headers=[]
for i in MasterArray.keys():
    for j,k in enumerate(MasterArray[i][0]):
        Headers.append(i+'_'+str(j))
RowFeaturesDataFrame = pd.DataFrame(FeaturesArray,columns=Headers)
toc = time.perf_counter()
tictoc = toc - tic
print(f'Tiempo de creaci n de la matriz {tictoc:0.4f}[s]')
print(art.separator)

print('* PASO 6: Importe del Modelo')
tic = time.perf_counter()
joblib_file = "SVM_model.pkl"
modelo = joblib.load(joblib_file)
toc = time.perf_counter()
tictoc = toc - tic
print(f'Tiempo de importacion del modelo {tictoc:0.4f}[s]')
print(art.separator)

print('* PASO 7: Predicci n')

```

```

tic = time.perf_counter()
FeaturesDataFrame = FeaturesDataFrame.append(RowFeaturesDataFrame,
    ignore_index=True)
OneSlicePredict = pd.DataFrame(scale(FeaturesDataFrame)).tail(n=1)
prediccion = modelo.predict(OneSlicePredict)
print(' La predicci n con SVM es: ', int(prediccion))
toc = time.perf_counter()
dispositivos = ['Licuadora', 'Bombilla', 'CFL', 'Cargador de Celular', '
    Computador de Escritorio', 'Ventilador', 'Nevera', 'Secador de Cabello',
    , 'L mpara Hal gena', 'Plancha', 'L mpara LED', 'Cargador de Laptop',
    , 'Sandwichera', 'TRC TV']
for i in range(len(dispositivos)):
    if prediccion[0] == i:
        resultado = dispositivos[i]
print(' El dispositivo detectado es', resultado)
tictoc = toc - tic
print(f'Tiempo de predicci n {tictoc:0.4f}[s]')
print(art.separator)

print('* PASO 8: Conexion a la Pagina Web')
tic = time.perf_counter()
cred = credentials.Certificate('firestore-nilmprototype-firebase-adminsdk
    -78hnb-5a249053b4.json')
firebase_admin.initialize_app(cred)
db = firestore.client()
main_coll = db.collection('nilm-prototype')
BlenderDoc = main_coll.document('blender')
BulbDoc = main_coll.document('bulb')
CFLDoc = main_coll.document('cfl')
Cellphone_ChargerDoc = main_coll.document('cellphone charger')
Desktop_PCDoc = main_coll.document('desktop pc')
FanDoc = main_coll.document('fan')
FridgeDoc = main_coll.document('fridge')
HairdryerDoc = main_coll.document('hairdryer')
Halogen_LampDoc = main_coll.document('halogen lamp')
IronDoc = main_coll.document('iron')
Laptop_ChargerDoc = main_coll.document('laptop charger')
TVDoc = main_coll.document('tv')
Led_LampDoc = main_coll.document('led lamp')
SandwichDoc = main_coll.document('sandwich maker')
Docs = [BlenderDoc, BulbDoc, CFLDoc, Cellphone_ChargerDoc, Desktop_PCDoc,
    FanDoc, FridgeDoc, HairdryerDoc, Halogen_LampDoc, IronDoc, Led_LampDoc,
    Laptop_ChargerDoc, SandwichDoc, TVDoc]
for i in range(len(Docs)):
    if prediccion[0] == i:
        Docs[i].update({'banderaCon':True, 'banderaDes':False, 'copPotencia
            ':PowerCalc, 'potencia':PowerCalc, 'tiempo':firestore.SERVER_TIMESTAMP
        })
toc = time.perf_counter()
tictoc = toc - tic
print(f'Tiempo de subida a la BD {tictoc:0.4f}[s]')
print(art.separator)

```
