PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado

Desarrollo de un enfoque de medición de energía eléctrica no intrusiva orientado a plataformas de bajo costo aplicado a usuarios finales utilizando técnicas de aprendizaje de máquinas

Tesis para obtener el grado académico de Maestro en Informática

que presenta:

José Luis Bruno Gutiérrez

Asesor:

Dr. César Armando Beltrán Castañón

Lima, 2025

Informe de Similitud

Yo, César Armando Beltrán Castañón, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis titulada(o) Desarrollo de un Enfoque de Medición de Energía Eléctrica no Intrusiva Orientado a Plataformas de Bajo costo Aplicado a Usuarios Finales Utilizando Técnicas de Aprendizaje de Máquina, de el autor José Luis Bruno Gutiérrez, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 10%. Así lo consigna el reporte de similitud emitido por el software Turnitin el 25/04/2025.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de investigación, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha:

Lima, 25 de Abril de 2025.

<table>
<tr><td colspan="2">Apellidos y nombres del asesor / de la asesora:
Beltrán Castañón, César Armando</td></tr>
<tr><td>DNI:
29561260</td><td rowspan="2">Firma
  </td></tr>
<tr><td>ORCID:
https://orcid.org/0000-0002-0173-4140</td></tr>
</table>

RESÚMEN

La creciente demanda de energía eléctrica junto con la necesidad de mitigar los efectos del uso de combustibles fósiles, exige el desarrollo de sistemas eficientes de monitoreo y medición del consumo energético. Este trabajo aborda el problema de identificar el consumo de cargas eléctricas en entornos domésticos a partir de datos agregados, con el objetivo de concientizar a los usuarios finales y promover la reducción del consumo de energía.

El principal objetivo fue diseñar y evaluar un sistema de medición eléctrica no intrusivo basado en técnicas de Aprendizaje de Máquinas, utilizando datos de potencia muestreados en baja frecuencia proporcionado en la base de datos REED (Residential Energy Efficiency Dataset). Se aplicaron ventanas temporales para extraer características y patrones de uso, lo que permitió entrenar y comparar varios algoritmos, incluyendo Random Forest, SVM, K-Nearest Neighbors, Gradient Boosting, sobre datos desbalanceados multiclase. Claves para el éxito del enfoque fueron el uso de la Codificación Binaria para representar el estado del sistema y la técnica de Centroides de Potencia para reducir la complejidad de los estados.

Los modelos se evaluaron utilizando la métrica F1-Score, los resultados mostraron que Random Forest alcanzó una puntuación de 0.89, destacándose en precisión. Extra Trees, con una mediana de 0.86, se posicionó como la mejor opción para entornos de recursos limitados. En clases con pocas instancias, SVM mostró un buen desempeño con un promedio de 0.74, mientras que KNN alcanzó un promedio de 0.72, ofreciendo una alternativa viable.

El estudio demuestra la viabilidad del monitoreo no intrusivo de cargas (Non-Intrusive Load Monitoring) mediante el enfoque propuesto, resaltando la efectividad de algoritmos basados en árboles de decisión para la desagregación. Además, se verificó la factibilidad de implementar este enfoque en dispositivos de recursos limitados como una Raspberry Pi, abriendo oportunidades para su aplicación en sistemas domóticos inteligentes.

ABSTRACT

The increasing demand for electricity, along with the need to mitigate the effects of fossil fuel usage, requires the development of efficient systems for monitoring and measuring energy consumption. This study addresses the problem of identifying individual appliance consumption in residential settings using aggregated data, with the aim of raising user awareness and promoting energy reduction.

The main objective was to design and evaluate a non-intrusive electrical measurement system based on Machine Learning techniques, using low-frequency power data from the REED (Residential Energy Efficiency Dataset). Temporal windows were applied to extract features and usage patterns, enabling the training and comparison of several algorithms, including Random Forest, SVM, K-Nearest Neighbors, and Gradient Boosting, on imbalanced multiclass data. Key elements to the success of the approach included the use of Binary Encoding to represent system states and the Power Centroid technique to reduce state complexity.

Models were evaluated using the F1-Score metric. Results showed that Random Forest achieved a score of 0.89, standing out in terms of precision. Extra Trees, with a median of 0.86, emerged as the most suitable option for resource-constrained environments. For classes with fewer instances, SVM performed well with an average score of 0.74, while KNN reached 0.72, offering a viable alternative.

The study demonstrates the feasibility of Non-Intrusive Load Monitoring (NILM) through the proposed approach, highlighting the effectiveness of tree-based algorithms for energy disaggregation. Furthermore, the implementation of this method on low-resource devices such as a Raspberry Pi was found to be feasible, opening opportunities for application in smart home systems.

ÍNDICE

LISTA DE TABLAS

LISTA DE FIGURAS

CAPITULO I

PLANTEAMIENTO DEL PROYECTO

1.1 Introducción

El mundo está experimentado un fuerte cambio climático cuyos efectos se observan en la alteración del hábitat natural, elevación del nivel del mar, aumento de fenómenos meteorológicos severos, sequías, entre otros. Entre las principales causas se encuentra la emisión de gases de invernadero provenientes de las actividades humanas; de ellas un componente importante es el uso de combustibles fósil [1]. Este cambio en el clima genera una preocupación a nivel mundial por lo que se busca atacar las causas de éste fenómeno. En países como Estados Unidos el 80% de la energía eléctrica proviene de restos fósiles [2] y en el Perú un 79.6% de la energía producida tiene su origen en fuentes derivadas del petróleo y el carbón [3] las que contribuyen con el incremento de las emisiones de gases de invernadero y en consecuencia aceleran el proceso de calentamiento global. Por éste motivo se están desarrollado políticas a nivel global para frenar éste fenómeno, por ejemplo, alentando el diseño, la formulación y la implementación de programas y proyectos relacionados al uso eficiente de la energía. Una gran parte de la energía eléctrica generada en el Perú se destina a los usuarios residenciales, esto subraya la necesidad de implementar medidas para fomentar la conciencia sobre el uso eficiente de la energía en este sector. Está demostrado que si el usuario es consiente y está atento a los consumos dentro de su vivienda se logra una reducción sustancial de hasta un 50% de la demanda de energía, lo que contribuye a atacar el problema de las emisiones de gases de invernadero por generación de energía eléctrica.

Este proyecto de tesis tiene como objetivo identificar técnicas de aprendizaje automático para su aplicación en sistemas de medición inteligente de energía eléctrica de bajo costo, dirigidos al usuario final. El propósito es proporcionar información clara y continua, permitiendo que los usuarios participen activamente en el control de su consumo energético.

1.2 Definición del problema

1.2.1 Situación Actual

Los sistemas de medición de energía eléctrica pueden dividirse en dos grupos, aparatos de simple medición de tipo analógico en los cuales no hay capacidad de almacenamiento de información y aparatos de medición digitales donde es posible medir varios parámetros eléctricos simultáneamente y almacenarlos para su posterior análisis. La medición de energía eléctrica comercial se lleva a cabo mediante el uso de un medidor o contador de energía y permite determinar el costo de la energía total que el usuario consume de acuerdo a las políticas de precio aplicables al sector. En todos los casos éstos equipos tienen la capacidad de realizar mediciones precisas según el punto de conexión, es decir si se conecta a la entrada de un circuito con varios aparatos eléctricos, medirá los parámetros totales o agregados de todas las

cargas. Si el objetivo es determinar la potencia desagregada, es decir los parámetros eléctricos de cada aparato en particular se deben hacer mediciones a la entrada de dicho aparato lo que resulta en un proceso tedioso y costoso por el uso de varios equipos de medición.

1.2.2 Situación Deseada

Actualmente hay investigaciones para el desarrollo de sistemas de medición inteligente que desagregue la carga eléctrica a partir de mediciones totales, a ésta técnica se le denomina Medición No Intrusiva y fue propuesto por Hart en el año 1992 [4]. Los sistemas de medición no intrusivos toman lecturas de tensiones, corrientes y potencias totales a la entrada del circuito eléctrico y con ésta información utiliza diversos algoritmos para determinar el consumo de cada aparato por medio de la identificación de alguna característica de la señal medida [5].

El uso de estos nuevos medidores permitirá obtener información que oriente tanto al usuario como al suministrador en la toma de decisiones con el objetivo de optimizar el consumo de energía eléctrica. Sin embargo, hay desafíos tecnológicos que se deben superar tanto a nivel de software como en hardware. En esta investigación se trabajará con bases de datos de consumos de energía eléctrica agregada para identificar el/los algoritmos de aprendizaje de máquinas que presente mejores ventajas en el diseño de un medidor inteligente.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar y evaluar un enfoque para la medición de energía eléctrica no intrusiva mediante la desagregación de cargas eléctricas basado en firmas de potencia promedio, utilizando técnicas de aprendizaje de máquinas y orientado a plataformas de bajo costo.

1.3.2 Objetivos Específicos

1.3.2.1 Objetivo Específico 1 (OE1)

Realizar una revisión literaria para identificar las técnicas de Aprendizaje de Máquina que pueden ser aplicados a los sistemas de medición no intrusiva.

1.3.2.2 Objetivo Específico 2 (OE2)

Desarrollar un software prototipo que implemente la técnica de Aprendizaje de Máquina para la desagregación de energía eléctrica.

1.3.2.3 Objetivo Específico 3 (OE3)

Realizar una comparativa de varias técnicas de Aprendizaje de Máquina, con miras a identificar aquellas que ofrecen mejores resultados según el enfoque planteado.

1.4 Resultados Esperados

En el Tabla 1 se presentan los resultados esperados propuestos para cada objetivo específico.

Objetivos	Resultados	
OE1	RE1	Identificación de las características y requerimientos de los algoritmos de ML aplicados a la identificación de cargas eléctricas.
	RE2	Selección y análisis de la base de datos de energía eléctrica.
OE2	RE3	Identificación y evaluación de las características de consumo de las cargas eléctrica.
	RE4	Desarrollo y evaluación del componente de extracción de características.
	RE5	Propuesta de arquitectura e integración de componentes
OE3	RE6	Evaluación de los modelos ensayados

Tabla 1. Resultados esperados por objetivo

1.5 Alcance

Este proyecto de investigación aplicada, en el área de la inteligencia artificial trata del diseño, ejecución y validación de un modelo de aprendizaje automático para la identificación de cargas eléctricas individuales a partir de datos agregados, inicialmente será ejecutado en una PC con conexión en red para luego ser desplegado en una plataforma electrónica con recursos limitados. El objetivo central es identificar el/los algoritmos que brinden mejores prestaciones según el enfoque planteado.

Para la realización de este proyecto, se ha identificado las limitaciones y alcance que hacen viable su desarrollo en el transcurso del periodo académico definido por la universidad.

El alcance está sujeto a las siguientes consideraciones:

- Utilizar un problema reducido de ambiente controlado y datos almacenados. No se trabajará con datos en tiempo real.
- Se seleccionará una limitada cantidad de aparatos electrodomésticos de un conjunto definido por la muestra.
- Se dará como válida la información de las bases de datos seleccionadas para el desarrollo.

1.6 Limitaciones

Se han identificado las siguientes restricciones:

- Solo se cuenta con datos de potencia eléctrica en baja frecuencia, se considera la tensión constante. Esto influye en la precisión de la identificación de la carga y la potencia que absorbe el dispositivo.
- No se tiene información específica de cada carga eléctrica de la muestra para identificar de manera precisa la potencia de la carga.

- La capacidad de procesamiento de la información está restringido a la capacidad de ejecución del hardware del proyecto.

1.7 Riesgos

- Puede ser difícil la identificación de aparatos de bajo consumo de energía debido a la resolución de la señal muestreada.
- Se puede confundir la identificación de aparatos electrónicos que tienen características constructivas y de funcionamiento similares.
- Disponibilidad de recursos económicos.

1.8 Justificativa

A continuación, se presentan los motivos de selección del tema de investigación, la motivación del proyecto y los beneficios que nos impulsan a culminar con éxito esta iniciativa.

- La creciente tendencia mundial hacia el uso de redes eléctricas inteligentes (Smart Grid) ha impulsado el desarrollo de los aparatos de medición inteligente. En este contexto, la necesidad de una medición detallada del consumo eléctrico de los usuarios, que beneficia al sistema, motivó la realización de esta investigación.
- Promover la reducción del consumo excesivo de energía eléctrica causado por el uso ineficiente de los aparatos eléctricos, contribuyendo directamente a mejorar la economía familiar
- Aportar a las investigaciones en el área de energía eléctrica, desarrollando técnicas que promuevan su uso eficiente y ayuden a mitigar el cambio climático.
- Contribuir al desarrollo del área de Electricidad en la universidad mediante la aplicación de técnicas de Inteligencia Artificial.

Los aparatos de medición electrónica son de uso masivo y necesario para la toma de decisiones a nivel técnico-económico–social, una información detallada del consumo por cada carga eléctrica ayuda a gestionar la demanda de la energía eléctrica tanto a nivel del usuario que puede controlar sus gastos como a nivel del proveedor que puede identificar de manera precisa los requerimientos de sus clientes. Más allá de esto con la información recabada y manteniendo la privacidad se puede orientar el uso de los datos a mejorar otros aspectos a nivel social como la seguridad y el confort de los usuarios. Así mismo, las técnicas de medición inteligente pueden ser aplicadas para otros ámbitos de desarrollo como la industria de producción, el transporte, la salud entre otros, en consecuencia, su aplicación es ilimitada.

CAPITULO II

ESTADO DEL ARTE

2.1 Marco General del Sistema de Medición no Intrusivo (NILM)

La medición eléctrica puede ser realizada por dos métodos: Medición Intrusiva y Medición No intrusiva [4] [6].

El método de medición Intrusivo requiere la intervención en cada parte del sistema eléctrico con el objetivo de conectar un sensor individual a cada aparato según se muestra en la Figura 1.

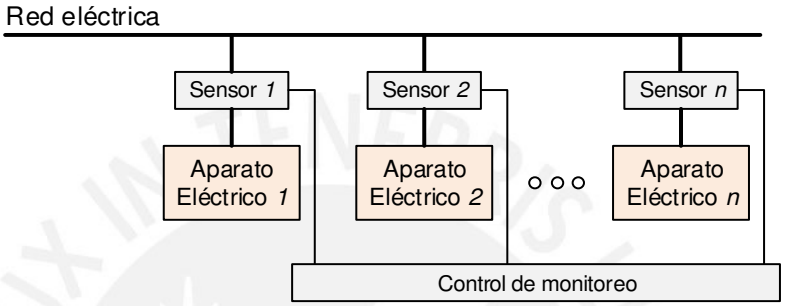


Figura 1. Sistema de medición instrusivo

El método de medición No Intrusivo sólo necesita la instalación de un instrumento medidor aguas arriba del sistema eléctrico, posteriormente los datos obtenidos son analizados para determinar la naturaleza y el consumo de energía de cada aparato individual. [5]

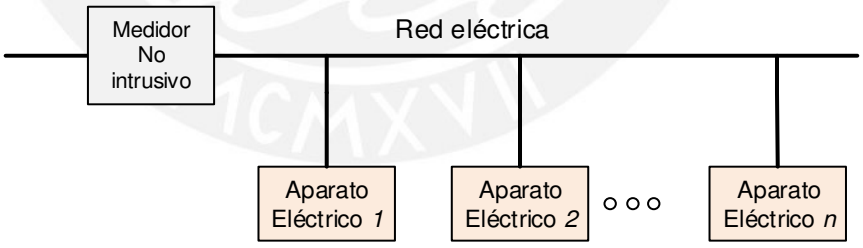


Figura 2. Sistema de medición no instrusivo

Después de la propuesta de Hart en 1992 [4] las investigaciones crecieron muy rápidamente y se desarrollaron algoritmos de desagregación a partir del reconocimiento de patrones y el aprendizaje automático. La Figura 3 muestra las partes básicas que forma el sistema NILM. [7]

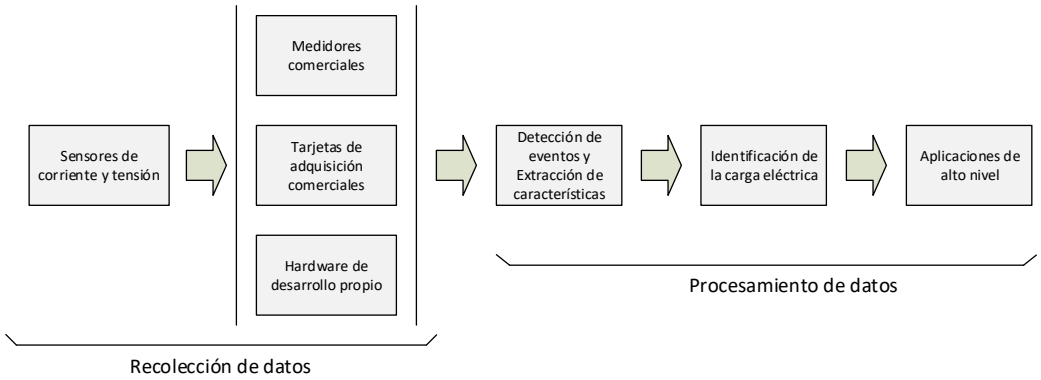


Figura 3. Partes de un sistema de medición no intrusivo

El proceso inicia con la obtención del consumo de energía eléctrica realizado en baja o alta frecuencia [8]. A continuación, es necesario identificar los eventos en la señal eléctrica, un evento está definido como un cambio en la señal, en este caso puede ser en tensión, corriente, potencia u otro. Para lograr la mayor precisión en la determinación de la carga eléctrica es necesario detectar las magnitudes y momentos exactos en los que ocurren los eventos. Con los eventos identificados se utilizan enfoques de estado estable [7], estado transitorio [5] y otros para identificar la carga individual y su consumo de energía, haciendo uso de la firma del dispositivo. La firma es una combinación de las características eléctricas que permitirá identificarlo dentro del dato agregado. Finalmente, utilizando las características extraídas y los datos etiquetados se realiza la clasificación identificando su estado de funcionamiento y su consumo de energía.

La identificación de la carga eléctrica es un problema de clasificación y optimización de cierta complejidad debido a las características y modos de funcionamiento variados de los aparatos eléctricos. Las técnicas utilizadas pueden ser con enfoque estadístico y de aprendizaje de máquinas.

2.2 Problema de Desagregación

La potencia total de una instalación eléctrica $P(t)$ debido a varios dispositivos eléctricos se expresa según la ecuación 1 [9]:

$$P(t) = P_{ruido} + \sum_{i=1}^{N} p_i(t), \quad t \in \{1, T\} \qquad (1)$$

Dónde T es el período de la onda eléctrica, p_i es la potencia de cada aparato y P_{ruido} es la potencia de la señal no deseada, definida como ruido eléctrico introducido durante la medición.

El objetivo de la desagregación es dividir la potencia total eléctrica adquirida por el medidor inteligente, en potencias parciales que representan el consumo de los dispositivos que lo componen. Una definición dada por Batra [10] es:

"Dada una secuencia discreta de lecturas de potencia agregadas observadas:

$x_t = x_1, x_2, \ldots x_T$, *determinar la secuencia de demandas de potencia del aparato eléctrico* $w_t^n = w_1^n, w_2^n, \ldots . w_T^n$ *donde* n *es uno de los N aparatos. Alternativamente este problema se puede representar como la determinación de los estados del aparato:* $z_t^n = z_1^n, z_2^n, \ldots . z_T^n$ *si se conoce un mapeo entre estados y demandas de potencia. Cada estado del aparato corresponde a una operación de consumo de energía aproximadamente constante (por ejemplo, "encendido", "apagado" o "en espera") y t representa una de las T medidas de tiempo discreto"*.

Las Figuras 4 y 5 presentan un ejemplo del proceso NILM como un problema de desagregación.

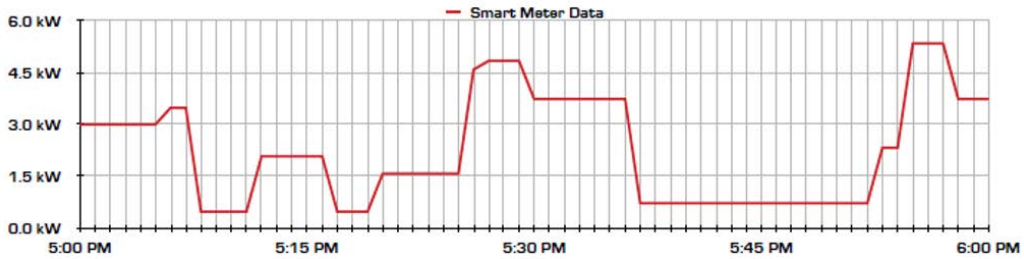


Figura 4. Datos agregados de potencia eléctrica. [11]

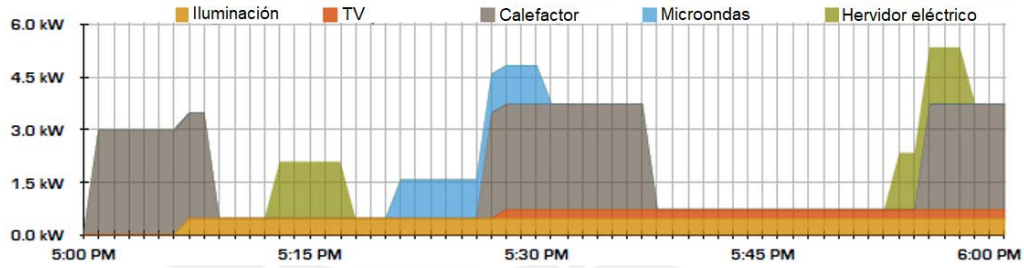


Figura 5. Potencia eléctrica desagregada, se observa que la carga total es la suma de las potencias individuales. [11]

2.3 Modelos de Aparatos Eléctricos

Los algoritmos de desagregación NILM dependen en gran medida de las diferentes firmas eléctricas, las cuales son definidas por el funcionamiento de cada aparato eléctrico. De acuerdo a Hart [4] se pueden considerar los siguientes tipos de aparatos:

a. Aparatos Eléctricos de Dos Estados

   Trabajan únicamente en encendido y apagado, tienen consumo fijo de potencia (Figura 6a), por ejemplo, las lámparas de iluminación resistiva, tostadoras, hervidores de agua entre otros.

b. Aparatos Eléctricos de Estado Finito (FSM)

   Son aquellos aparatos que tienen varios estados de funcionamiento durante su ciclo de trabajo, por ejemplo, un refrigerador que presenta 3 estados: apagado,

encendido y descongelado (Figura 6b) o las máquinas lavadoras (Figura 6d). Cada uno de estos estados tendrá un consumo de energía específico y un patrón de transición repetible que permitirá su identificación en los datos agregados.

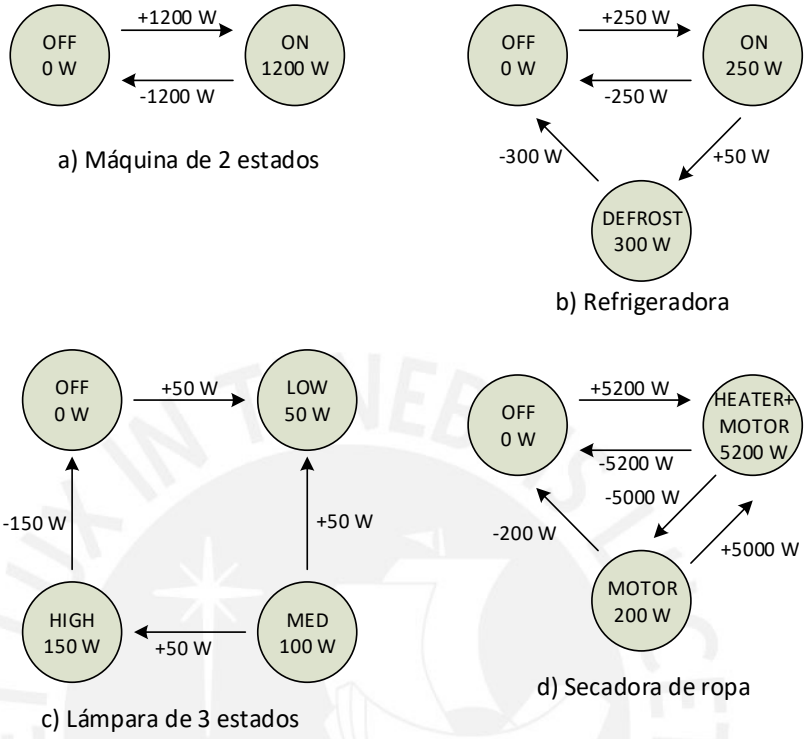


Figura 6. Ciclo de trabajo de aparatos eléctricos [4]

c. Aparatos eléctricos continuamente variables (CVM)

Aquellos aparatos eléctricos que tienen condiciones de operación variable como se muestra en la Figura 7 y no se puede definir el nivel de potencia con precisión. Por ejemplo, un taladro o un atenuador de luz (dimmer). Son aparatos difíciles de identificar a partir de datos agregados, ya que no tienen un cambio de estado claro y pueden exhibir características de una amplia gama de funciones.

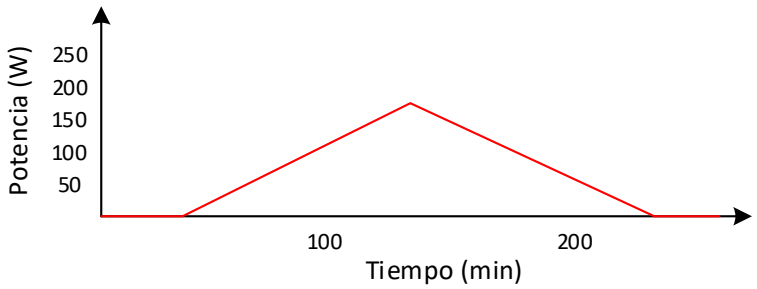


Figura 7. Máquina continuamente variable [4]

d. Aparatos eléctricos de funcionamiento permanente

Aquellos aparatos se encuentran encendidos de manera permanente. Generalmente son de bajo consumo de energía como las alarmas de seguridad,

detectores de humo, despertadores entre otros. Son difíciles de desagregar, ya que no tienen eventos.

2.4 Adquisición de Datos

El sistema que recoge los datos de la red eléctrica es denominado AMI (Advanced Metering Infraestructure). Está compuesto por medidores inteligentes que tienen la capacidad de registrar, analizar y transmitir la información eléctrica. De acuerdo a la frecuencia de muestreo de la señal se pueden dividir en dos categorías [8]:

a. Sistemas de baja frecuencia

   Utiliza frecuencias de muestreo menores a 1Hz. Son los recomendados en función a la escalabilidad y el costo, pero tiene funcionalidad limitada. En estas condiciones solo se pueden extraer características "macroscópicas" de la señal eléctrica.

b. Sistemas de alta frecuencia

   Las frecuencias de muestreo se encuentran en el orden de kHz a MHz. En estas condiciones se puede obtener formas de onda de corriente, tensión, armónicos y transitorios de arranque, esta condición de operación nos permite obtener las características llamadas "microscópicas".

Los medidores eléctricos de uso comercial operan en bajas frecuencias, mientras que para obtener mediciones de alta frecuencia se debe utilizar sistemas embebidos dedicados o ad-hoc, los cuales ofrecen ventajas adicionales, pero con costo elevado [7]. Actualmente hay un amplio uso de aparatos de medición inteligentes (Smart Meter) con tendencia a reemplazar a los medidores analógicos de baja frecuencia.

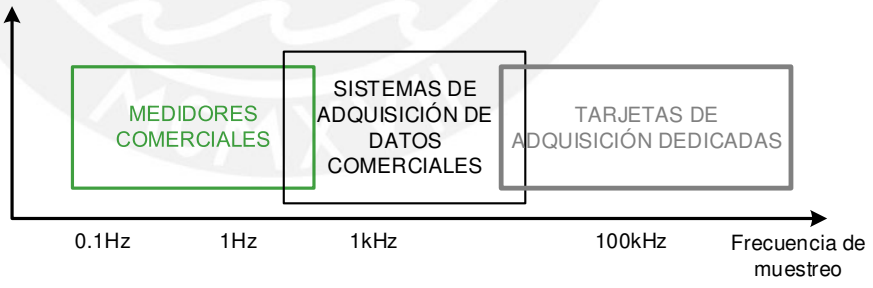


Figura 8. Equipos de medición electrónica y su desempeño con la frecuencia [8]

Según Kolter y Matthew [12] dos aspectos a tener en cuenta para la selección o diseño del sistema embebido de adquisición son la frecuencia de muestreo y la arquitectura del hardware implementado.

2.5 Características de los Aparatos Eléctricos Orientados a la Desagregación de Energía

Con el objetivo de identificar al aparato eléctrico dentro de una carga agregada primero se debe seleccionar las mejores características para las firmas de éste dispositivo, Le [13] establece que las características de ingeniería son inherentes a su funcionamiento, por ejemplo, potencia, corriente o energía. Una identificación más precisa se encuentra en Zoha [6], que los clasifica en características de estado estable, transitorio y no tradicional.

2.5.1 Características de Estado Estable

Hart [4] propone el uso de la potencia Activa o Real (P) y la Reactiva (Q) para identificar el encendido o apagado del aparato eléctrico, así también se puede utilizar el Factor de Potencia (FP). El FP es la relación entre la potencia Real P (W) y la potencia total S (VA) y está asociado a la diferencia de fase entre la tensión y la corriente. Norford [14] y Farinaccio [15] consideran que los aparatos de gran consumo de potencia se identifican rápidamente, sin embargo, existe un problema cuando dos o más aparatos que tienen el mismo consumo de energía funcionan al mismo tiempo como se muestra en la Figura 9 [6], en este caso la propuesta de Drenker [16] es utilizar la relación entre la potencia real y reactiva.

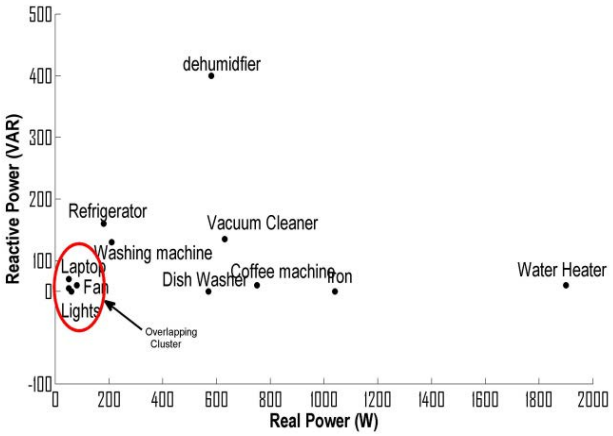


Figura 9. Distribución de potencia activa (W) y reactiva (VAR) [6]

Ruzelli [17] propone el análisis de las formas de onda de corriente (I) y la tensión (V) a través de algunas de sus características como los valores pico, eficaz o RMS, así como el factor de potencia (FP). Los valores característicos antes mencionados no tienen buen rendimiento en el caso de aparatos de varios estados de funcionamiento. Si la muestra de señal es suficientemente amplia es posible trabajar con la Transformada de Fourier y Wavelets como lo proponen Su, Lian y Chang [18]. Lam [19] propone utilizar las trayectorias V-I para categorizar a los aparatos eléctricos, en éste caso se utilizan valores normalizados de tensión y corriente para identificar diferentes trayectorias V-I como se muestra en la Figura 10, lo que conduce a la obtención de una

taxonomía de aparatos eléctricos demostrando que el enfoque es más efectivo que los enfoques existentes basados en mediciones de potencia.

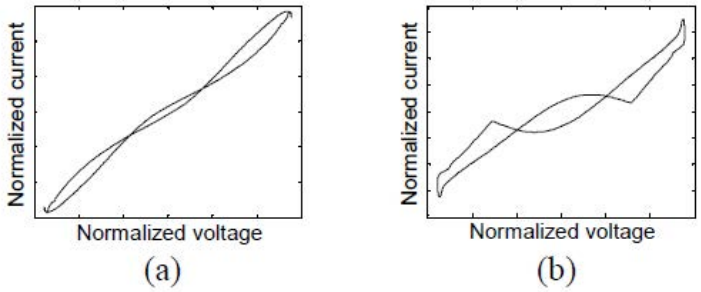


Figura 10. Trayectorias V-I para a) Máquina de aire acondicionado y b) Horno microondas [19]

2.5.2 Características de Estado Transitorio

Cuando el aparato eléctrico cambia de un estado estable a otro se presenta un estado transitorio, en este lapso de tiempo los parámetros eléctricos y magnéticos cambian rápidamente y su identificación permite una desagregación precisa, pero para obtenerlos se requieren muestreos de alta velocidad.

Chang [20] y Cole [21] propusieron usar la energía desarrollada durante los transitorios de funcionamiento para discriminar aparatos eléctricos. También es posible utilizar la transformada de Fourier o la transformada de Wavelets asociado a las potencias real P y reactiva Q. [22] [23].

Patel y Gupta [24] utilizaron los ruidos eléctricos de tensión y las radiaciones electromagnética (EMI) en alta frecuencia que se presentan en las líneas eléctricas debido al funcionamiento de aparatos con fuentes de alimentación conmutadas. Utilizando estas emisiones es posible discriminar cargas del mismo tipo ubicadas en zonas distintas de la instalación eléctrica, pero los ruidos e interferencias dependen de la topología de la instalación eléctrica.

2.5.3 Características no Tradicionales

Con el objetivo de mejorar la desagregación se puede hacer uso de características no eléctricas, por ejemplo, incluir el horario, la estación del año, el ruido audible, la ocupación de la red de datos entre otros, según se muestra en [25] y [26].

2.6 Extracción de Características de los Datos.

Con los datos adquiridos comienza el proceso de identificación o clasificación de la carga eléctrica. Esta etapa es compleja porque el sistema eléctrico es dinámico y con una gran cantidad de equipos conectados a ella con varios estados operativos en cada caso.

La clasificación o reconocimiento utiliza técnicas informáticas que pueden ser agrupadas según la propuesta de Zhuang [8]:

a. Optimización.
b. Probabilidad.
c. Aprendizaje automático y reconocimiento de patrones.
d. Aprendizaje profundo.

a. Optimización.

   El objetivo es maximizar la coincidencia de las características extraídas de los aparatos eléctricos en funcionamiento a los datos previamente almacenados y debidamente identificados. Se utilizan diferentes métodos para realizar la búsqueda combinatoria, Egarter [27] propuso una solución utilizando un algoritmo evolutivo para la identificación de aparatos de tipo ON/OFF sin considerar cargas reales y con el inconveniente del elevado tiempo ejecución y su naturaleza estocástica, por otro lado, Suzuki [28] utilizó datos de corriente de alta frecuencia y aplico Programación Entera en la desagregación, posteriormente Bhotto [29] mejoró la precisión de la desagregación incluyendo diagramas de estado, filtros y refinamiento basado en programación lineal, en estos casos el modelamiento matemático y la complejidad que deriva del incremento de aparatos eléctricos es una desventaja. Otras propuestas son la programación híbrida y la programación cuadrática restringida de enteros segmentados [30].

   Estos métodos suelen obtener una alta precisión durante la identificación, sin embargo, demandan de una gran capacidad de procesamiento.

b. Probabilidad

   Wong [31] considera que la señal eléctrica de un electrodoméstico es la salida de un sistema estocástico, y propone mantener una representación de estado del sistema completo, en lugar de tratar los eventos individuales, entonces es factible aplicar enfoques probabilísticos como los Modelos Ocultos de Markov (HMM) [32]. La aplicación implica cierta complejidad debido al incremento de las cargas eléctricas que redunda en un incremento de la dimensionalidad. Hay varias investigaciones enfocadas en dar solución a éste desafío, entre ellas tenemos los que propone variantes de HMM. Kolter [33], Parsons [34], Johnson [35] y Makonin [11] han utilizado variantes de HMM factoriales. Parson propone establecer modelos genéricos de electrodomésticos y utilizar una extensión del algoritmo de Viterbi para identificar un dispositivo de la carga total agregada. Johnson utilizó un Modelo Semi Oculto de Markov (HSMM) para incluir la duración del estado de la carga. Zia [36] utilizó los datos de las sub mediciones para construir modelos HMM de cada electrodoméstico para luego combinarlas y hacer un HMM más grande. Zeifman [37] utilizó los niveles de potencia y su duración para crear distribuciones

de probabilidad de energía y tiempo de uso. Makonin [11] analizó los datos de sub mediciones de la carga y creó una función de masa de probabilidad para cada equipo, obteniendo de este modo una HMM de súper estado que fue desagregado mediante el algoritmo de Viterbi disperso. Se observa que es factible aplicar HMM en entornos de pocos dispositivos y con datos debidamente identificados para estimar las probabilidades de transición, sin embargo, los sistemas eléctricos son ruidosos y con alta correlación entre dispositivos lo que representa una desventaja para su aplicación.

c. Aprendizaje Automático y Reconocimiento de Patrones

El aprendizaje automático con el reconocimiento de patrones se utiliza ampliamente en sistemas NILM. Zoha [6] presenta un análisis de varios enfoques comunes, luego de un exhaustivo análisis presentan una tabla de comparaciones de los algoritmos aplicados (Tabla 2). Así mismo varios investigadores han sugerido combinaciones de métodos de aprendizaje con técnicas de reducción de dimensiones para lograr mayor eficiencia en la desagregación, por ejemplo, Figueredo [38] aplicó Support Vector Machines (SVM), y K-Nearest Neighbors (k-NN) junto con las firmas de potencia y factor de potencia de los estados estacionarios para desagregar las cargas eléctricas. Por su parte Giri [39] entrenó varios clasificadores comunes de aprendizaje automático y logró determinar que KNN tiene un mejor desempeño entre las alternativas exploradas, para esto realizó medición de campo electromagnético (EMF) y Técnicas de Reducción de la Dimensionalidad (PCA). Luego de sus investigaciones Srinivasan [40] concluyó que las Máquinas de Vectores de Soporte (SVM) muestran un buen rendimiento en la clasificación de dispositivos, especialmente cuando se utiliza firmas de ondas armónicas y características de baja frecuencia.

Es posible representar los estados de operación de los aparatos eléctricos en ventanas temporales, en esa dirección Lin [41] utiliza una secuencia de observaciones de consumos de potencia representadas mediante etiquetas y de este modo entrenó modelos KNN y SVM. EL uso de modelos híbridos es sustentado por Lai [42] que propone la combinación de SVM/GMM en el que GMM se utiliza para describir la distribución de las formas de ondas actuales a fin de encontrar similitudes de potencia; mientras que el SVM realiza la clasificación de las características de potencia extraídas para reconocer las operaciones de las cargas eléctricas. Las conclusiones de Dimas [43], muestran que, para el caso de pocos datos, el algoritmo KNN demuestra ser más efectivo mientras que en un escenario de gran cantidad de datos el algoritmo del Árbol de Decisión es superior. También hay aportes importantes en el trabajo de Moreno [44] al incluir el método de Componentes Principales (PCA) y Random Forest (RF) en datos de potencia activa en muestreo de baja frecuencia, concluye con una eficacia global superior

al 92%. Por su parte Berrettini [45] hizo un estudio comparativo de clasificadores para la desagregación utilizando una caracterización de la carga eléctrica basada en el teorema de la conservación de potencia, luego propuso tres modelos de aprendizaje automático para validar la correcta identificación de la carga, k-NN, Máquina de Vectores de Soporte (SVM) y Bosque Aleatorio (RF). Concluye que el algoritmo RF presentó el mejor desempeño, con respecto al cálculo computacional y la precisión.

Otros trabajos en los que se utilizaron el clasificador Naïve Bayes [39], [46] y [47], Campos Aleatorios Condicionales de Cadena Lineal (CRF) para manejar cargas multiestado [48], demuestran la factibilidad de utilizar estos enfoques. Ruano [7] hace un aporte importante cuando lista una recopilación de investigaciones con análisis exhaustivo de las principales características de las técnicas NILM seleccionadas con tasas de muestreo muy bajas, bajas y medias, que luego da a conocer mediante tablas detalladas donde resalta las principales contribuciones.

Learning Algorithm	Features St [a]/Tr [b]	Accuracy %	Training S [c]/U [d]	Online/ Offline	Scalability	Appliance Types
SVM [11,17,33,54]	B [e]	75–98	S	Online	Yes	I, II, III & IV
Bayes [12,50,54]	St	80–99	S	B	No	I & II
HMM [49,59,60]	St	75–95	B	Offline	No	I & II
Neural Networks [17,37,61]	B	80–97	S	Online	Yes	I & II & III
KNN [6,9,62]	B	70–90	S	B	Yes	I & II
Optimization [7,18,20,35]	St	60–97	S	Offline	No	I & II

[a] Steady-State [b] Transient [c] Supervised [d] Unsupervised [e] Both.

Tabla 2. Comparación de desagregación de carga eléctrica [6]

El uso de Aprendizaje Automático brinda ventajas con respecto a la flexibilidad y generalización de datos complejos, sin embargo, debemos considerar la alta carga computacional y la dificultad en la interpretabilidad, de algunos algoritmos, por lo que se debe identificar de manera adecuada la correcta aplicación de esta técnica.

d. Aprendizaje Profundo

Es una poderosa herramienta para la desagregación de cargas eléctricas como demuestran las siguientes investigaciones. Zhou [49] presenta uno de los primeros trabajos NILM donde se utiliza un modelo de reconocimiento de patrones con redes neuronales (NNPR), en su trabajo utiliza pocos parámetros de medición con a una tasa de muestreo baja, realiza simulaciones y presenta una evaluación del desempeño, concluye que para los aparatos que tienen potencia variable no proporciona un método de detección eficaz. Sirojan [50] aplica redes neuronales profundas mediante la combinación de redes neuronales convolucionales y codificadores automáticos variacionales durante el proceso de desagregación de la carga eléctrica. Arnav Kundu [51] construye un modelo de red neuronal para

dos electrodomésticos típicos y plantea la detección de características con el objetivo de identificar el cambio de estado, para esto utiliza Redes Neuronales Convolucionales (CNN) seguido del uso de Redes Neuronales Recurrentes (RNN) de Memoria a Largo y Corto Plazo (LSTM), demuestra que la detección del cambio de estado es afectada cuando se tienen cargas de perfiles similares, del mismo modo si los aparatos eléctricos tienen funcionamiento irregular la red neuronal resulta ser un predictor débil. Miao Weiwei [52] proponen detectar los eventos de activación y desactivación de la carga eléctrica mediante el método de descomposición SVD de empuje por inmersión, a partir de esto obtener datos característicos para la base de datos de entrenamiento de la red neuronal de identificación de carga. Ruttagorn Prasertlux [53] utilizó datos recopilados de 10 tipos de equipos con una mejora con respecto al tiempo de entrenamiento de otros algoritmos NILM de aprendizaje profundo. Se desarrolla un perfil de carga inclusive identificando perfiles de carga de equipos que tienen un nivel de potencia similar. Las ventajas son amplias, en contraposición se debe cumplir con ciertos requisitos como; disponer de datos masivos y etiquetados, contar con una buena capacidad de computo debido a que suelen requerir uso de GPUs o TPUs, por otro lado, la dificultad en la interpretación y una infraestructura especializada condicionan su aplicación.

CAPITULO III

MARCO CONCEPTUAL PARA LA IDENTIFICACIÓN DE LA CARGA ELÉCTRICA

Tomando como referencia las investigaciones revisadas, se ajustaron los requerimientos y se concluye en:

- Utilizar las firmas de potencia de estado estacionario.
- Utilizar datos de baja frecuencia.
- Reducir la carga computacional.
- Aplicar técnicas de aprendizaje de máquinas que muestren el mejor desempeño.
- Utilizar técnicas para reducir la dimensionalidad.

Se define un marco conceptual centrado en la propuesta.

3.1 Característica Eléctrica para Definir la Onda de Potencia Eléctrica

Se utilizan los siguientes parámetros eléctricos:

3.1.1 Potencia Media o Promedio (μ_P). A partir de la potencia muestreada se obtiene la potencia media o potencia promedio que se utiliza para caracterizar el consumo de energía neto durante la ventana de tiempo, su unidad son los vatios (W). Matemáticamente de define según la ecuación 2.

$$\mu_P = \frac{1}{T}\int_{t_0}^{t_0+T} p(t)dt \quad T: Período \tag{2}$$

3.1.2 Potencia Pico (P_m). Está definido como el mayor valor de la potencia muestreada en el lapso de una ventana de tiempo y corresponde a la mayor potencia a la que opera el aparato eléctrico, se define en vatios (W).

3.1.3 Energía Eléctrica (E). Es la capacidad de realizar un trabajo. Corresponde a la energía promedio que consume un determinado electrodoméstico durante un período de tiempo. Se define según la ecuación (3) y su unidad es vatio-segundo (W-s).

$$E = P \times t \tag{3}$$

$P\ es\ potencia\ eléctrica\ (W)$

$t: tiempo\ (s)$

3.2 La Carga Eléctrica, la Firma de la Carga Eléctrica (LS) y la Carga Agregada

3.2.1 Aparato Eléctrico. Es un dispositivo que utiliza energía eléctrica para su funcionamiento, una Carga Eléctrica está formada por uno o más aparatos eléctricos, y se identifica por su consumo de energía en vatios (W).

3.2.2 La Firma de la Carga Eléctrica (LS). Todo aparato eléctrico presenta características particulares que lo identifican de un conjunto, a esto denominamos la Firma Eléctrica. Liang [54] define la firma de la carga eléctrica como la unidad básica y el medio más importante para desarrollar el sistema de medición no intrusivo (NILM). En forma matemática se define como $\psi_i(t)$ en la ecuación (4)

$$\Psi_i(t) = \left\{f_{i,1}(\vec{v},t), f_{i,2}(\vec{v},t), \ldots., f_{i,M}(\vec{v},t) | \Delta t = T\right\} \tag{4}$$

Donde $\vec{v}$ es un vector de medidas eléctricas básicas, t es tiempo; $f(\vec{v},t)$ es una característica extraída de $\vec{v}$; M es el número total de características; y T es el intervalo de muestreo de $\vec{v}$. Es importante definir el período de muestreo T para diferenciar tipos de firmas de carga.

3.2.3 La Carga Agregada. La firma de la carga debe estar acompañada por la definición de la carga eléctrica compuesta (CL) o carga agregada. Liang [54] define como el comportamiento de más de un aparato eléctrico que operan simultáneamente y se describe como $\Omega(\mathrm{t})$ en la ecuación (5):

$$\Omega(t) = \left\{ \sum_{i=1}^{R} \Psi_i(t) | \Delta t = T \right\} \qquad (5)$$

Donde R es el número total de aparatos que funcionan simultáneamente.

La carga agregada tiene un comportamiento más complejo debido a la combinación de los consumos de potencias de las cargas individuales en funcionamiento, como se muestra en la Figura 11, sin embargo, esta información es más accesible.

Dado que cada máquina tiene un patrón de consumo definido por sus características intrínsecas es posible identificar comportamientos a los que llamamos estados de los aparatos eléctricos.

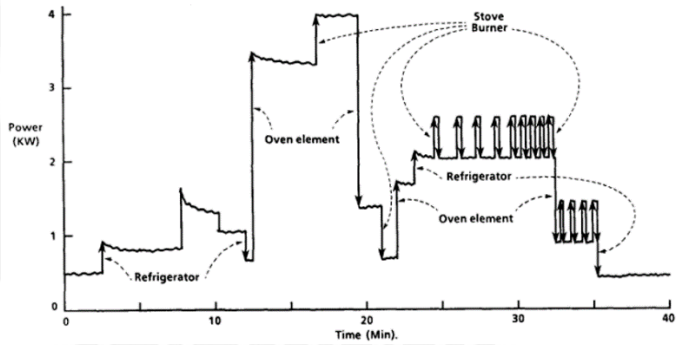


Figura 11. Cambios en la potencia total debido al consumo individual de la carga eléctrica [4].

3.3 Herramientas Estadísticas Para Cuantificar el Error

Para la estimación de los errores tenemos varias herramientas:

3.3.1 Error Absoluto Medio (MAE): Calcula el promedio de las diferencias absolutas entre las predicciones y los valores de referencia. Cuanto más bajo sea el valor de MAE, mejor será el modelo en sus predicciones. La fórmula es:

$$MAE = \sum_{i=1}^{N} \frac{|\varphi_i - \varphi_{iobs}|}{N} \qquad (6)$$

Donde:

φ_i es el valor pronosticado para la celda i

φ_{iobs} es el valor observado para la celda i

N es el número de valores analizados

Una desventaja de MAE es que depende de la escala y comparar valores entre diferentes escalas puede llevar a una interpretación errónea. También MAE otorga el mismo peso a todas las observaciones, independientemente de su importancia o impacto y en aplicaciones de energía ciertos datos pueden ser más críticos que otros.

3.3.2 Error Cuadrático Medio (MSE): Calcula la media de los errores al cuadrado entre las predicciones y los valores de referencia. En este caso se penaliza más a las predicciones que están lejos de la referencia. La fórmula es:

$$MSE = \sum_{i=1}^{N} \frac{(\varphi_i - \varphi_{iobs})^2}{N} \qquad (7)$$

3.3.3 Raíz del Error Cuadrático Medio (RMSE): Es la raíz cuadrada del MSE, tiene la ventaja de estar en la misma escala que los valores de referencia y es una medida de la concentración de los datos

La fórmula es:

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(\varphi_i - \varphi_{iobs})^2}{N}} \qquad (8)$$

Es de fácil interpretación, pero sensible a errores grandes y por ello a valores atípicos haciendo un sesgo hacia arriba.

3.3.4 Error Porcentual Absoluto Medio (MAPE): El MAPE es útil cuando se trabaja con datos porcentuales o valores relativos. Calcula el promedio de los errores porcentuales entre las predicciones y los valores de referencia. La fórmula es:

$$MAPE = \frac{1}{N} \times \sum_{i=1}^{N} \left| \frac{\varphi_i - \varphi_{iobs}}{\varphi_{iobs}} \right| \times 100 \qquad (9)$$

φ_i es el valor pronosticado para la celda i

φ_{iobs} es el valor observado para la celda i

N es el número de valores analizados

Es particularmente útil para comprender la magnitud de los errores en relación con los valores reales. Una limitación de MAPE es que deja de estar definido cuando el valor real es cero.

3.4 Herramientas Estadísticas Para Describir a la Onda de Potencia

Con el objetivo de diferenciar una onda eléctrica de otra, utilizaremos las características que derivan de su comportamiento en el tiempo, Chowdhury [55] propone varias características, entre ellas tenemos:

3.4.1 Media ($\bar{x}$): Brinda información cuantitativa de la ubicación central de los datos. La media es simplemente un promedio numérico.

$$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n} \qquad (10)$$

Donde n es la cantidad de datos de la muestra

3.4.2 Desviación Media Absoluta (MAD): Es una medida de dispersión, se utiliza para calcular la variabilidad de un conjunto de datos. MAD calcula el promedio de la distancia ente cada par de puntos de datos real y ajustado. Se define como:

$$MAD = \frac{\sum_{i=1}^{n}|x_i - \bar{x}|}{n} \qquad (11)$$

Donde: x_i son los valores individuales en el conjunto de datos.

$\bar{x}$ es la media aritmética de los datos.

n es el número de valores en el conjunto de datos.

3.4.3 Varianza (s^2): Utilizado como una medida de la distribución de los puntos en el conjunto de datos sobre el valor medio. La varianza se puede expresar matemáticamente como:

$$s^2 = \frac{(x_i - \bar{x})^2}{n-1} \qquad (12)$$

Donde: s^2 es la varianza, x_i denota los datos de la distribución, $\bar{x}$ es el valor medio del total de datos y n denota el número total de datos o el tamaño de la distribución. La varianza es considera en la teoría inferencial, sin embargo la Desviación Estándar (s) se utiliza más en las aplicaciones.

$$s = \sqrt{s^2} \qquad (13)$$

Ambas medidas reflejan el mismo concepto en la variabilidad de la medición, pero la desviación estándar de la muestra mide la variabilidad en unidades lineales.

3.5 Algoritmos de Aprendizaje de Máquinas

Enfocado en los algoritmos, Tom Mitchell [56] define “aprendizaje” como “*programas informáticos que mejoran su desempeño en alguna tarea a través de la experiencia*”, en ese sentido un algoritmo de aprendizaje de máquinas aprende de los datos, lo que permite mejorar su rendimiento de manera automática.

Los algoritmos de Aprendizaje de Máquinas pueden clasificarse en: Aprendizaje Supervisado, No Supervisado y por Refuerzo. A continuación, se lista una breve referencia de los más representativos, para mayor detalle puede referirse a una amplia variedad de referencia bibliográfica disponible.

El Aprendizaje Supervisado busca explicar una o más variables llamadas dependientes en término de otras variables independientes. Donde las salidas correctas del modelo predictivo actúan como “supervisores” de los resultados obtenidos durante el aprendizaje.

3.5.1 Logistic Regression

Se utiliza en problemas de clasificación. La regresión logística estima la probabilidad de que una instancia pertenezca a una de dos clases aplicando una función logística (también conocida como Sigmoide) según la relación mostrada:

$$P_{(y=1/X)} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}} \qquad (14)$$

Donde:

$P_{(y=1/X)}$ es la probabilidad de que la instancia pertenezca a la clase positiva de las características X.

e es la base del logaritmo natural.

$\beta_0, \beta_1, \ldots \beta_n$ son los coeficientes del modelo que se aprenden durante el entrenamiento.

$x_1, x_2, \ldots, x_n$ son las características de la instancia.

La regresión logística utiliza técnicas de optimización para ajustar los coeficientes β de manera que se maximice la verosimilitud de los datos observados.

3.5.2 K-Nearest Neighbors

Conocido como los K-Vecinos más Cercanos. Es un sencillo algoritmo no paramétrico que se basa en la idea intuitiva de que los ejemplos similares tienden a pertenecer a la misma clase o tener valores de salida similares. K-NN clasifica una nueva instancia calculando la clase más común entre los "k" ejemplos más cercanos a esta instancia en el espacio de características, "k" que debe ser especificado antes de la ejecución del algoritmo. Se basa en el cálculo de la distancia entre la nueva instancia y cada una de las instancias en el conjunto de datos de entrenamiento. La distancia más comúnmente utilizada es la distancia Euclidiana.

3.5.3 Naive Bayes Classifier

Este clasificador se basa en el teorema de Bayes con una suposición "naive" (ingenua) de independencia condicional entre las características. Se establece la relación entre la probabilidad condicional de un evento y la probabilidad marginal de los eventos relacionados.

Este clasificador estima la probabilidad de pertenencia a una clase dada una instancia x mediante la fórmula del teorema de Bayes:

$$P_{(y/x)} = \frac{P_{(x/y)} \times P_{(y)}}{P_{(x)}} \qquad (15)$$

Donde:

$P_{(y/x)}$ es la probabilidad condicional de que la instancia y pertenezca a la clase x

$P_{(x/y)}$ es la probabilidad condicional de observar la instancia x dada la clase y

$P_{(y)}$ es la probabilidad marginal de la clase y

$P_{(x)}$ es la probabilidad marginal de la clase x

Una vez que se han calculado las probabilidades condicionales para cada clase, se selecciona la clase con la probabilidad condicional más alta como la predicción del clasificador.

3.5.4 Decision Tree

Se basa en una estructura de árbol donde cada nodo interno representa una característica (atributo), cada borde (rama) representa una regla de decisión basada

en esa característica, y cada hoja representa la etiqueta de clasificación o el valor de regresión resultante.

El proceso implica dividir repetidamente el conjunto de datos de entrenamiento en subconjuntos más pequeños basados en las características que mejor separan las clases. Esto se hace de manera recursiva hasta alcanzar la condición de parada dada. Los algoritmos más utilizados son CART (Classification and Regression Trees) y el algoritmo ID3 (Iterative Dichotomiser 3). Entre los criterios de división puede ser la ganancia de información, la ganancia de Gini o la reducción de error cuadrático.

3.5.5 Support Vector Machine (SVM)

Es un algoritmo utilizado principalmente para clasificación. La separación de los conjuntos de datos se realiza mediante la búsqueda de un hiperplano óptimo que maximiza el margen entre las clases. Un hiperplano en un espacio de "n" dimensiones es un subespacio de dimensión "n-1". Las instancias más cercanas a este hiperplano son conocidas como los vectores de soporte para definir el hiperplano, lo que hace que sea eficiente en la clasificación de conjuntos de datos de gran tamaño.

SVM utiliza una técnica llamada "kernel trick" para mapear los datos a un espacio de características de mayor dimensión donde la separación lineal puede ser posible. Los kernels más comunes utilizados son el kernel lineal, el kernel polinomial y el kernel radial (RBF).

3.5.6 Multilayer Perceptron (MLP)

Es un tipo de red neuronal artificial con múltiples capas de nodos (neuronas) interconectados entre sí. Está compuesto por una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa está formada por un conjunto de neuronas que están conectadas a las neuronas de la capa anterior y/o siguiente mediante conexiones ponderadas. Es un modelo de aprendizaje supervisado utilizado para problemas de clasificación y regresión. Durante el entrenamiento, el modelo ajusta los pesos de las conexiones entre las neuronas para minimizar una función de pérdida que mide la diferencia entre las predicciones del modelo y las etiquetas verdaderas de los datos de entrenamiento. Este proceso se lleva a cabo mediante algoritmos de optimización como el descenso del gradiente estocástico (SGD) o sus variantes.

El MLP es capaz de aprender funciones altamente no lineales y realizar tareas de clasificación y regresión en conjuntos de datos complejos.

Los Modelos Ensamblados conocidos como métodos de conjunto, combinan múltiples modelos base (“modelos débiles" o "learners") para mejorar el rendimiento predictivo en comparación con un solo modelo. Múltiples modelos pueden reducir el sesgo y la varianza, lo que lleva a una mejor generalización y predicciones más precisas.

3.5.7 Bootstrap Aggregating (Bagging). Se entrena un conjunto de modelos base utilizando subconjuntos aleatorios (con reemplazo) del conjunto de datos de entrenamiento. Cada modelo base puede ser de cualquier tipo, como árboles de decisión, SVM o cualquier otro algoritmo de aprendizaje supervisado. Después de entrenar todos los modelos base, las predicciones de cada modelo se combinan utilizando votación mayoritaria (en clasificación) o promedio (en regresión) para producir la predicción final. Es efectivo para reducir la varianza y prevenir el sobreajuste, especialmente en modelos propensos a este problema, como los árboles de decisión.

3.5.8 Gradient Boosting Machines (GBM). Se utiliza para problemas de regresión y clasificación. Se basa en el concepto de Boosting, que consiste en la construcción secuencial de modelos débiles, donde cada modelo intenta corregir los errores de los modelos anteriores.

Los modelos base generalmente son árboles de decisión poco profundos, de manera aditiva. El algoritmo tiene como objetivo reducir gradualmente los residuos utilizando el gradiente descendente y minimizando la función de pérdida. Inicia con un modelo base simple que predice una estimación inicial del objetivo, luego calcula los residuos entre las predicciones y la etiqueta, para ajustar un nuevo árbol y minimizar la función de pérdida de los residuos. El proceso se repite hasta alcanzar un criterio de parada predefinido, como un número máximo de iteraciones o la convergencia del modelo.
Es ampliamente utilizado debido a su rendimiento excepcional.

3.5.9 Adaptive Boosting (AdaBoost)

Construye un clasificador fuerte a partir de varios clasificadores débiles siguiendo el siguiente procedimiento. Entrena un clasificador débil utilizando el conjunto de datos de entrenamiento asignando pesos iguales a todas las muestras de entrenamiento.
Luego de la clasificación, las muestras mal clasificadas reciben un peso mayor, mientras que las bien clasificadas reciben un peso menor. El proceso de asignación de pesos y construcción del clasificador final se realiza de manera iterativa, dando lugar a un clasificador fuerte que generalmente tiene un rendimiento mejor que los clasificadores individuales. AdaBoost es especialmente efectivo en conjuntos de datos desequilibrados y puede manejar tanto datos categóricos como numéricos.

3.5.10 Random Forest

Es una técnica de conjunto (ensemble) que se basa en la construcción de múltiples y diversos árboles de decisión y combina sus predicciones para obtener un resultado más robusto y preciso.

Esto introduce diversidad en los árboles individuales, lo que ayuda a reducir el sobreajuste y mejorar la generalización del modelo. Se realiza un muestreo bootstrap (muestreo con reemplazo) del conjunto de datos de entrenamiento. La predicción final de Random Forest se determina por votación mayoritaria entre los árboles

individuales. Es capaz de manejar eficazmente conjuntos de datos grandes y capturar relaciones no lineales y es menos propenso al sobreajuste.

En el aprendizaje No Supervisado los algoritmos se utilizan para encontrar patrones o estructuras intrínsecas en un conjunto de datos sin la presencia de etiquetas o respuestas predefinidas. Se realizan tareas de agrupamiento o clustering utilizando técnicas como K-Means o DBSCAN, también reducción de la dimensionalidad aplicando Análisis de componentes principales (PCA).

3.5.11 K-Means. Es un algoritmo de agrupamiento (clustering) que divide un conjunto de datos en k grupos o clusters de manera que los puntos de datos dentro de un mismo cluster sean similares entre sí y distintos de los puntos de datos en otros clusters

Se define de la siguiente forma. Sea un conjunto de datos D con n puntos $\boldsymbol{x}_i$ en un espacio $d - dimesional$, y dada la cantidad de grupos (clusters) deseados k, el objetivo de K-Means es dividir el conjunto de datos en k grupos, denotado como $C = \{C_1, \dots, C_K\}$ Además, para cada grupo C_i existe un punto que representa al grupo, siendo en este caso la media o centroide (μ_i)

$$\mu_i = \frac{1}{n_i} \sum_{x_j \in C_i} X_j \quad (16)$$

Donde $n_i = |C_i|$ es el número de puntos en el cluster C_i.

Además si los datos agregados es S_{agg} tal que:

$$S_{agg} = \bigcup_{k=1}^{K} C_k \text{ y} \quad (17)$$

$$C_k \cap C_l = \emptyset \text{ , para } k \neq l \quad (18)$$

Se muestra el pseudocódigo y el proceso de clustering – Kmeans.

Algoritmo K-means:

1. **Especificar** el número K de clústeres a asignar.
2. **Inicializar** aleatoriamente K centroides.
3. **Repetir**
4. **expectativa:** Asignar cada punto a su centroide más cercano.
5. **maximización:** Calcular el nuevo centroide (media) de cada grupo.
6. **hasta que** Las posiciones del centroide no cambien.

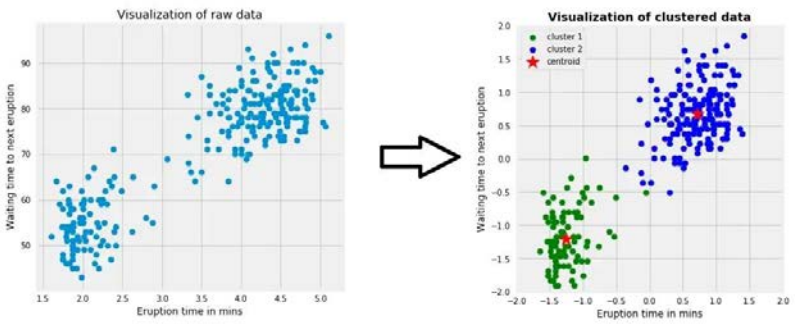


Figura 12. Proceso de Clustering [57]

3.6 Manejo de Datos Multiclase y Desbalanceado

La distribución desequilibrada de los datos no facilita la clasificación ya que puede provocar predicciones sesgadas del modelo, el problema se presenta cuando la clase minoritaria de interés, está sub representada ya que los algoritmos de aprendizaje tienden a estar sesgados hacia grupos con más instancias. En ese sentido se han propuesto métodos para mejorar el rendimiento en la clasificación. Abokadr [58] ilustra esta problemática en la Figura 13 y también propone soluciones potenciales para mitigar los efectos del desbalance.

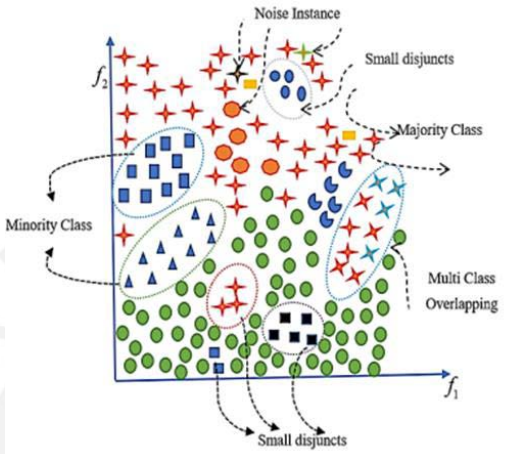


Figura 13. Desafíos de datos desbalanceados [58]

Submuestreo, reduciendo el tamaño de la clase mayoritaria eliminando muestras aleatoriamente de ella. Esta técnica es adecuada cuando la clase mayoritaria es amplia en comparación a la clase minoritaria y el conjunto total es grande.

Sobremuestreo, aumentando el tamaño de la clase minoritaria con muestras sintéticas cogeneradas. Esta técnica es adecuada cuando la clase minoritaria tiene pocas muestras en comparación a la clase mayoritaria y el conjunto total es relativamente pequeño. Un ejemplo es la técnica SMOTE.

También es posible utilizar técnicas de Aprendizaje sensible a los costos modificando el algoritmo para dar pesos diferenciados, utilizar métodos de conjunto combinando las predicciones de múltiples modelos para mejorar el rendimiento, y los métodos de aprendizaje profundo utilizando redes neuronales.

3.7 Métricas Aplicadas Para la Validación del Modelo

Son medidas cuantitativas utilizadas para evaluar el rendimiento y la calidad de un modelo de aprendizaje automático determinando la capacidad de generalización y predicción.

Entre las razones más importantes para su aplicación están la evaluación del rendimiento del modelo de una manera objetiva frente a datos nuevos, la comparación entre modelos utilizando las mismas métricas para identificar el de mejor desempeño en una tarea de predicción específica, la optimización de hiperparámetros mediante la opción de ajuste con el objetivo de mejorar el rendimiento del modelo y la detección de sobreajustes y subajustes.

Se considera *V*: Verdadero, *F*: Falso, *P*: Positivo y *N*: Negativo en las predicciones, para definir las siguientes métricas de validación:

3.7.1 Accuracy (Exactitud): Mide la proporción de instancias clasificadas correctamente entre el total de instancias. Es fácil de interpretar, recomendable si las clases están relativamente balanceadas ya que presenta limitaciones en problemas con clases desbalanceadas al haber muchos más datos de alguna de ellas.

$$Accuracy = \frac{VP+VN}{VP+VN+FN+FP} \quad (19)$$

3.7.2 Precisión (Precisión): Mide la proporción de instancias positivas predichas correctamente entre el total de instancias predichas como positivas. Se centra en la calidad de las predicciones positivas. Es especialmente importante cuando los falsos positivos son costosos o problemáticos.

$$Precision = \frac{VP}{VP+FP} \quad (20)$$

3.7.3 Recall (Recuperación o Sensibilidad): Mide la proporción de instancias positivas predichas correctamente entre el total de instancias positivas reales. Se centra en capturar la mayor cantidad posible de instancias positivas. Es crucial en situaciones donde los falsos negativos son costosos o problemáticos.

$$Recall = \frac{VP}{VP+FN} \quad (21)$$

3.7.4 F1-Score: Es la media harmónica de Precision y Recall. Es útil cuando hay un desequilibrio entre precision y recall ya que proporciona un equilibrio entre ambas buscando minimizar los falsos positivos y falsos negativos.

$$F_1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (22)$$

3.7.5 Área Bajo la Curva (ROC): Es una métrica que evalúa la capacidad de discriminación de un modelo de clasificación binaria en diferentes umbrales de decisión. La curva ROC (Receiver Operating Characteristic) representa la tasa de verdaderos positivos frente a la tasa de falsos positivos.

SEGUNDA PARTE: DISEÑO METODOLÓGICO Y RESULTADOS

CAPÍTULO IV

METODOLOGÍA

En este capítulo se describe la metodología empleada para llevar a cabo la investigación, que incluye el diseño y los métodos de análisis utilizados para identificar el algoritmo que presente mejores resultados en la medición no Intrusiva según el enfoque propuesto.

Se plantea los siguientes pasos:

- ✓ Análisis de los datos.
- ✓ Extracción de características de la carga eléctrica.
- ✓ Obtención del DataFrame.
- ✓ Evaluación de los algoritmos en la identificación de la carga eléctrica

4.1 Análisis de los Datos.

4.1.1 Análisis de Potencia en Aparatos con Múltiples Estados.

Según lo revisado en 2.3 los aparatos eléctricos tienen un comportamiento con varios sub estados de funcionamiento. Hay aparatos de encendido y apagado (ON/OFF), aparatos con comportamiento de estados finitos (FSM) y los que varían de manera permanente. Sin embargo, el funcionamiento de un aparato eléctrico no sólo contempla los estados de funcionamiento, también se debe analizar los estados transitorios propios de la máquina y aquellos que no dependen de ésta. Al analizar la onda temporal de la potencia en la base de datos REDD se observa que un aparato eléctrico de dos estados (ON/OFF) como la iluminación tiene una onda característica como se muestra en la Figura 14.

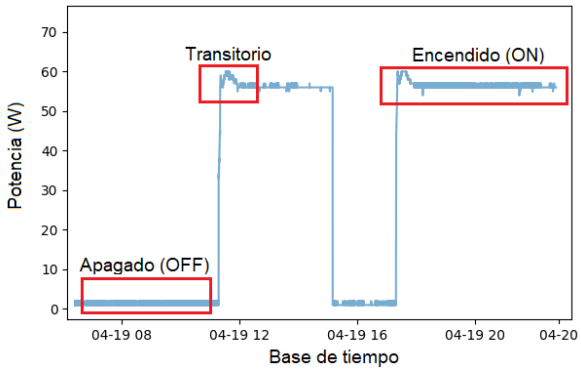


Figura 14. Máquina de dos estados

Se identifican dos estados bien definidos, sin embargo, se observan ligeros cambios sobre el nivel promedio a los que denominamos transitorios. Estos cambios son valores no característicos que se presentan debido a errores en la lectura o transitorios inherentes a su funcionamiento.

La Figura 15 muestra la onda de potencia de un refrigerador, y se observa la condición de operación estable y otros consumos no característicos debido a una condición particular de funcionamiento del aparato.

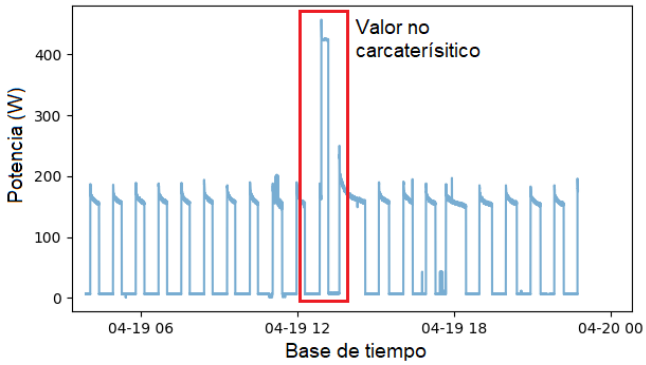


Figura 15. Consumo de una refrigeradora

El objetivo es estimar el consumo de energía a partir del estado estable, entonces los valores transitorios de corta duración no se toman en consideración porque no tienen mayor efecto sobre el consumo. Los aparatos como las lavadoras de ropa (máquinas de varios estados), tienen un modo de funcionamiento definido por ciclos como se muestra en la figura 16. Para cargas de éste tipo se observa que hay valores de potencia definidos según el ciclo de trabajo de la máquina

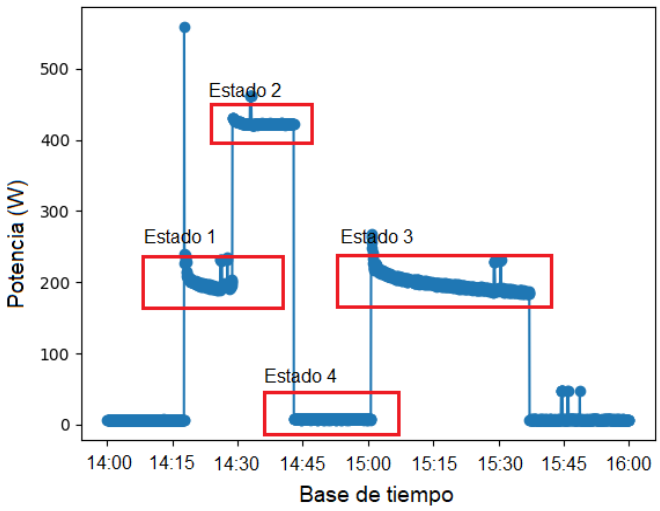


Figura 16. Consumo de una máquina lavadora

Los niveles de potencia pueden variar de un estado a otro, sin embargo, se identifican valores característicos que podemos utilizar como referencia de consumo. La Figura 17 muestra el consumo de un horno eléctrico, estos aparatos se caracterizan por ser cargas resistivas puras y generalmente llevan un control de temperatura por corte y activación. Se observa que es una máquina de dos estados con consumos de potencia bien definidos y con períodos inactivos relativamente largos.

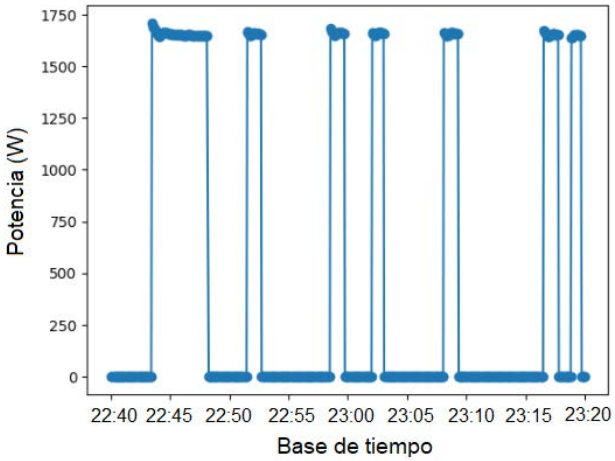


Figura 17. Consumo de un horno eléctrico

La Figura 18 muestra el consumo de un horno de microondas, estos aparatos eléctricos generalmente funcionan por cortos períodos de tiempo, pero con altos consumos de potencia.

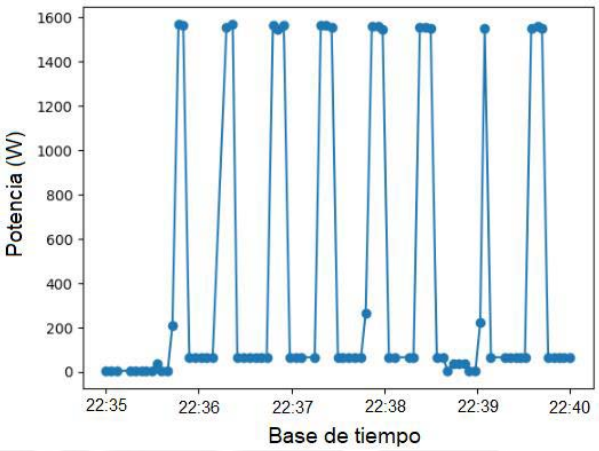


Figura 18. Consumo de un horno microondas

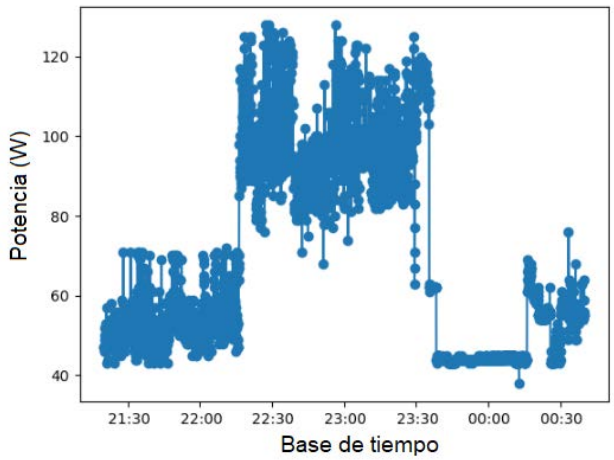


Figura 19. Consumo de un tomacorriente

La Figura 19 muestra la potencia que se demanda de un tomacorriente, estos consumos presentan una gran variación y no tienen un patrón definido ya que no se atribuye a ningún aparato eléctrico identificado.

4.1.2 Centroides de Potencia.

Como se vio en 4.1.1 las cargas eléctricas trabajando en estado estable tienen potencias de consumo con variaciones dentro de su potencia nominal en ese sub estado. Por ejemplo, la iluminación tiene una potencia de encendido, una lavadora puede tener cuatro valores característicos de potencia que corresponden a enjuague, lavado, secado y centrifugado. Mediante la determinación de las frecuencias de repetición de las potencias es posible determinar las potencias de los estados característicos como se muestra en la Figura 20.

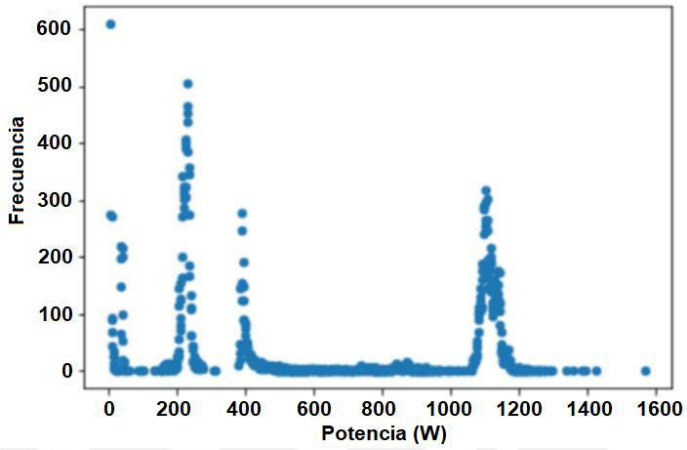


Figura 20: Potencias características de una carga eléctrica

Donde se identifican las potencias representativas centradas en 200W, 400W y 1100W, también se observan valores dispersos de baja potencia atribuidos al modo de operación de reposo o inoperativo, y otros de mayor valor relacionados con los transitorios de operación. Es factible aplicar técnicas de Clustering para identificar estos valores centrales, uno de estos es K-Means debido a su simplicidad y pocos parámetros de ajuste que brinda una eficiencia computacional.

4.1.3 Determinación de los Centroides usando KMeans.

Se pueden establecer agrupaciones de potencia definidos para cada sub estado, a los que denominaremos Centroides de potencia o Clúster de potencia y están centrados alrededor de la potencia promedio consumida en ese sub estado, como se muestra en La Figura 21.

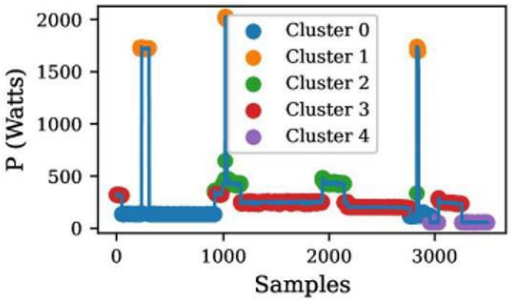


Figura 21. Muestra de la segmentación de datos en la señal de potencia eléctrica [71]

El clustering de potencias nos permite la eliminación de los ruidos o variaciones no significativas en las mediciones y al mismo tiempo permite identificar patrones de consumo en grupos de cargas. Trabajos que sustentan la discretización de la potencia en centroides o bloques de potencia son Makonin [11] y Puente [55]. Un ejemplo de la aplicación se muestra en la Figura 22.

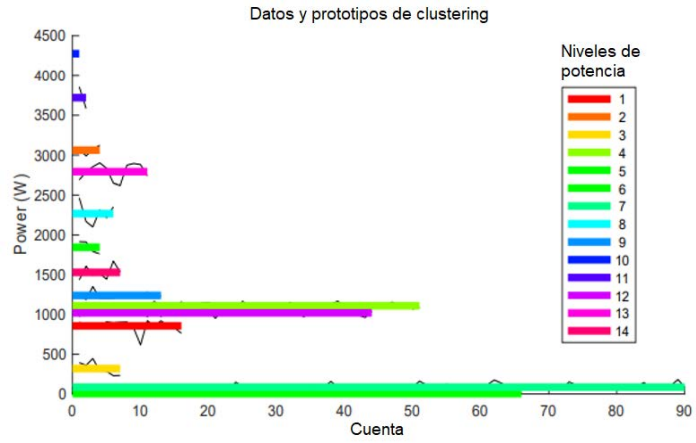


Figura 22. Clustering de potencia [60]

Para la aplicación de KMeans se debe tener en cuenta:

- ✓ El Número de clústeres (K).
- ✓ Elección del valor inicial.
- ✓ Minimizar los valores atípicos.
- ✓ Elección de medidas de distancia.

El número de clusters está directamente relacionado con la cantidad de estados estables del aparato eléctrico, se reconoce un estado estable porque hay una variación relativamente suave de los valores de potencia. Para determinar el número óptimo de clústeres se utiliza el "método del codo". El procedimiento es aplicar el algoritmo Kmeans y a continuación calcular la Inercia o varianza intraclúster, para graficarla junto al número de cluster, luego se identifica el "Codo" o punto en el que la disminución de la inercia se desacelera significativamente, este punto es el número óptimo de clústeres como se muestra en la Figura 23.

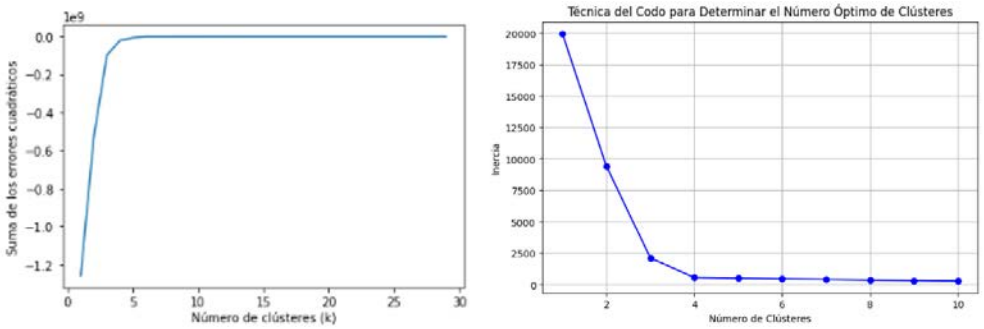


Figura 23. Aplicación del método de codo para determinar el número óptimo de clúster.

El objetivo es tener grupos con datos compactos y bien diferenciados entre ellos, por lo que la elección de la cantidad de grupos es uno de los aspectos más importantes.

Hay dos características que se deben considerar: Cohesión y separación, como se muestra en la Figura 24.

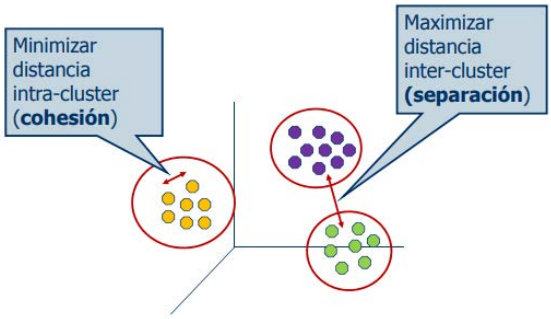


Figura 24. Cohesión y separación en Clustering

4.1.4 Análisis de los Centroides de Potencia.

Al aplicar el método Kmeans a los datos de potencia se identifican los centroides o potencias representativas como se muestra en la Tabla 3.

ID	Aparato eléctrico	C1 (W)	C2 (W)	C3 (W)	C4 (W)	C5 (W)
EQ0	Tomacorriente 3	152	1259	79	101	-
EQ1	Tomacorriente 4	0	118	405	1312	-
EQ2	Aparato electrónico 6	0	729	208	131	-
EQ3	Refrigerador 7	4	718	581	111	-
EQ4	Horno 10	0	2295	2497	2237	-
EQ5	Iluminación 11	0	2513	267	2739	-
EQ6	Lavadora 13	2	1718	124	391	-
EQ7	Iluminación 17	0	1289	1595	947	-
EQ8	Iluminación 19 (4 cluster)	130	2	1038	366	-
EQ8	Iluminación 19 (5 cluster)	130	2	911	1205	366

Tabla 3. Potencias centroides de las cargas eléctricas

Hay una relación directa entre el centroide de cada cluster (Ci) y el comportamiento de la onda temporal de potencia como se muestra en las Figuras 25, donde (a) es la curva del codo que establece el número mínimo de clusters, (b) muestra la ubicación de los centroides en la distribución de los datos y (c) identifica el estado estable aproximado en el desarrollo temporal de la onda de potencia. La aplicación

de la técnica del codo a la carga 6 en house 3 de los datos REDD sugiere la partición en dos clusters como se observa en la Figura 25-a, y es justificado mediante la agrupación de datos en potencias menores y mayores a 300W como se ve en la Figura 25-b. Debido a la dispersión de éstos grupos es convenientes tomar cuatro clusters cuyos centroides son las potencias representativas 0.9 W, 131.5 W, 208.9 W, 729 W que son identificados en la representación de la potencia instantánea como se muestra en la figura 25-c.

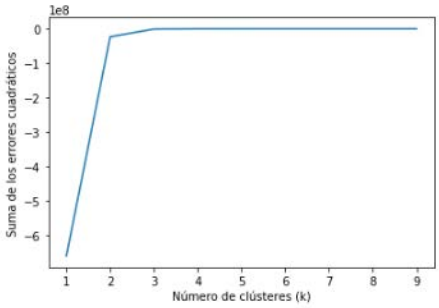


Figura 25 a) Gráfica de Codo para Aparato electrónico 6

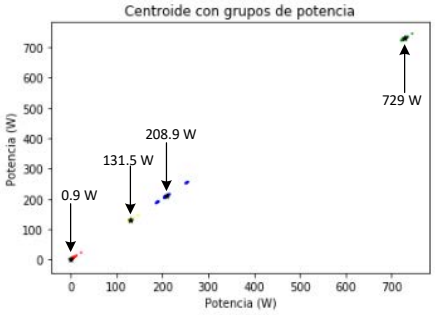
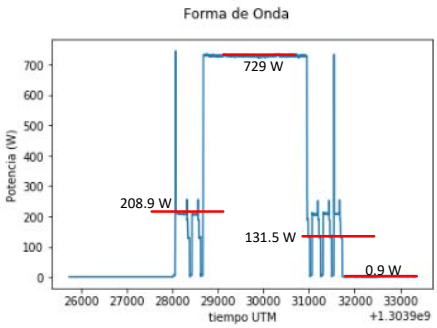


Figura 25 b) Gráfica de Centroides para el Aparato electrónico 6



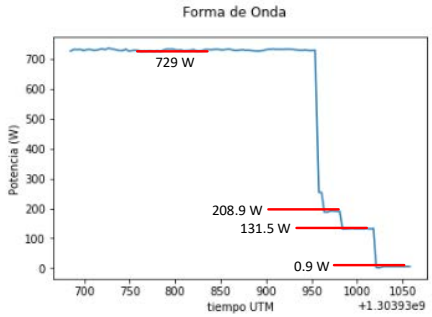


Figura 25 c) Gráfica de potencia en el tiempo para Aparato electrónico 6

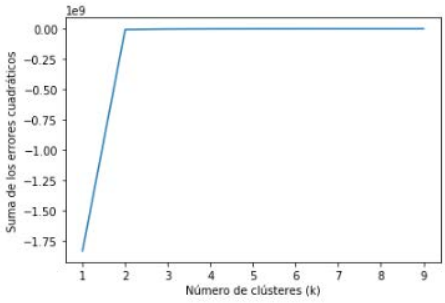


Figura 26 a) Gráfica de Codo para Lavadora 13

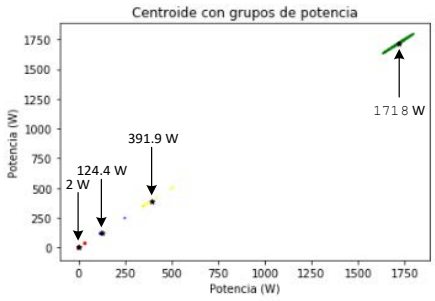


Figura 26 b) Gráfica de Centroides para Lavadora 13

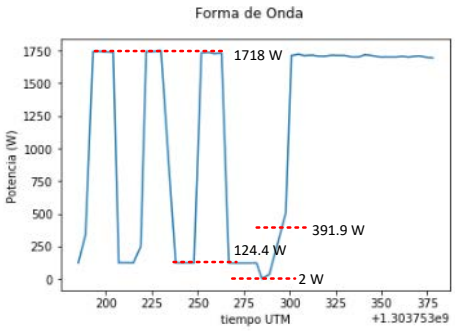


Figura 26 c) Gráfica de potencia en el tiempo para Lavadora 13

Por tanto, la determinación del número de cluster no sólo depende de la aplicación del método del codo, también es necesario tener una comprensión de los datos. Como ejemplo para la carga 13 de REED identificado como “Lavadora”, la curva de codo sugiere 2 clusters, sin embargo, observando la distribución de los datos es convenientes considerar 4 clusters como se muestra en la figura 26-b cuyos centroides son las potencias representativas 2 W, 1718 W, 124.4 W, 391.9 W. La Figura 26-c muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

4.1.5 Efecto de la Ventana de Tiempo en la Determinación de los Centroides de Potencia

La obtención de la energía eléctrica está relacionada con el tiempo de muestreo de la potencia según la ecuación:

$$Energía = Potencia\ X\ tiempo \qquad (23)$$

Dónde *Potencia* es la potencia instantánea medida y *tiempo* es la ventana de tiempo para la acumulación de la energía. La ventana de tiempo permite formar las instancias, y a partir de esta se obtienen los parámetros eléctricos como potencia media, desviación estándar, energía, que son necesarios para entrenar el modelo.

La elección del tamaño de la ventana está definida en base a los siguientes criterios:

a) Tiempo de Funcionamiento del Aparato Eléctrico. El tiempo de funcionamiento del aparato eléctrico define el consumo de energía. Algunos aparatos trabajan por períodos cortos de tiempo como los hornos de microondas, desde algunos segundos a varios minutos, otros como una máquina lavadora o una lámpara de iluminación tienen funcionamiento más prolongado que va de algunos minutos a varias horas, en consecuencia, la ventana de tiempo debe contener información suficiente para identificar características del funcionamiento de los aparatos eléctricos.

b) La unidad de tiempo para la facturación de la energía eléctrica es de 15 minutos que impone una restricción para la duración de la ventana de tiempo.

c) Una ventana de tiempo pequeña permite obtener mayor detalle respecto a las condiciones atípicas de funcionamiento de la máquina como por ejemplo las condiciones de arranque o transitorios.

4.1.6 Análisis de las Ventanas de Tiempo

El uso de ventanas de tiempo permite obtener tendencias y patrones de la energía eléctrica consumida, así también se convierte en la base de tiempo para la obtención de las variables del circuito. Establecemos el ancho de la ventana en función del detalle definido para el análisis, si las ventanas son pequeñas se pueden identificar funcionamientos inusuales de los aparatos como por ejemplo los transitorios, y si se incrementa, se obtendrá una visión más general del funcionamiento siendo más adecuada para la obtención del consumo de energía.

A continuación, se muestra el desarrollo de la potencia eléctrica y la energía consumida por una carga eléctrica en un tiempo 1200 segundos. La Figura 27 muestra los centroides de potencia obtenidos para diferentes tiempos de la ventana. La curva de color azul (●), muestra el consumo de potencia promedio instantáneo de la carga eléctrica. La curva roja (∗) muestra los valores de potencia asignados por el algoritmo de clustering que efectivamente son los centroides en esa ventana de tiempo. Se observa que, según se incrementa el tamaño de la ventana de 30 segundos a 150 segundos la gráfica se va suavizando. Considerando que el objetivo es determinar la energía eléctrica consumida, es más importante para el análisis el área bajo la curva de potencia. Con ventanas de 60 segundos, se emplea un único centroide para representar la potencia media, mientras que con ventanas de 90 segundos, se requieren dos centroides para el mismo intervalo de tiempo. Esto se justifica por la dependencia de la potencia media respecto al período de tiempo.

La Tabla 4 muestra los centroides de potencia que modelan a 9 aparatos eléctricos considerando una ventana de tiempo de 90 segundos y 4 potencias centrales identificadas mediante clustering, queda entendido del análisis anterior que habrá un sesgo o desviación con respecto al valor verdadero instantáneo.

ID	Aparato eléctrico	C1 (W)	C2 (W)	C3 (W)	C4 (W)
EQ0	Tomacorriente 3	146	86	772	474
EQ1	Tomacorriente 4	0	117	388	61
EQ2	Aparato electrónico 6	0	727	106	183
EQ3	Refrigerador 7	4	671	406	140
EQ4	Horno 10	0	1276	1935	812
EQ5	Iluminación 11	0	1318	2095	323
EQ6	Lavadora 13	2	1698	623	1144
EQ7	Iluminación 17	0	1297	454	907
EQ8	Iluminación 19	1	132	997	392

Tabla 4. Potencia Centroide para una ventana de tiempo de 90 segundos

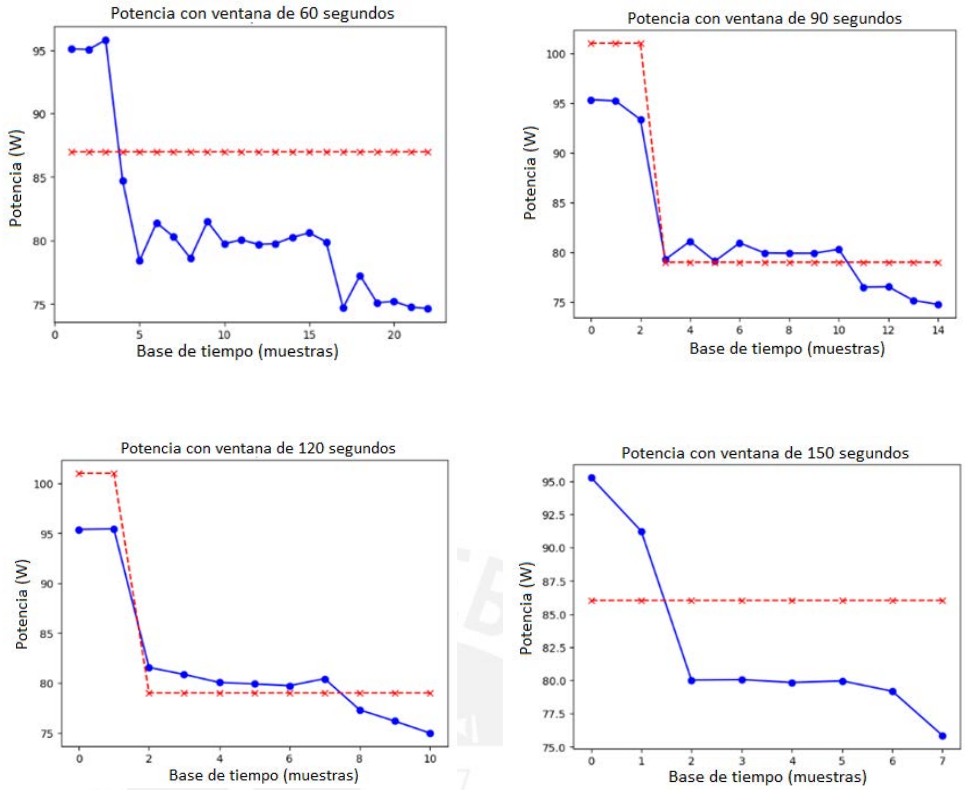


Figura 27. Detalle de la onda de potencia al cambiar el tamaño de la ventana de tiempo. Azul: potencias promedio instantánea (●). Rojo: Centroide de la onda (∗)

Usando los centroides para representar la potencia consumida por el aparato eléctrico se reduce significativamente la cantidad de estados de funcionamiento de la máquina, como se muestra en la Figura 28.

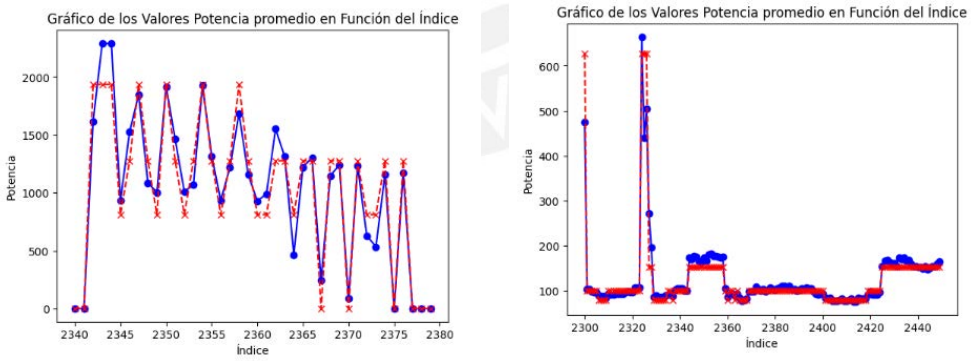


Figura 28. Datos ajustados a los valores del centroide: (●) onda de potencia instantánea, (∗) onda de potencia ajustada al centroide.

Es necesario verificar la capacidad del modelo para ajustar los valores verdaderos a los valores del centroide de potencia para estimar de manera confiable los consumos de potencia de la carga, si el modelo es bueno debe tener un sesgo bajo.

Las potencias instantáneas pueden tener infinitos valores, para minimizarlos se propone ajustarlos a un grupo reducido de potencias consumidas y para lograr este objetivo es necesario tener ciertas consideraciones como las que se indican:

- Utilizando algoritmos de clustering se definen niveles de potencia centrales, dado que las potencias instantáneas varían permanentemente se introducirán errores de ajuste.
- Los aparatos eléctricos pueden consumir niveles de potencia muy diferenciados, desde algunos vatios a cientos o miles de vatios, debido a esto los errores deben interpretarse de manera distinta para diferentes escalas, por ejemplo un aparato que consume 2W en la condición de operación A y 200W, en la condición de operación B, un error de 50% implica una potencia de 1W para el caso A y 100W en el caso B, claro está que 1W está comprendido dentro de la tolerancia de funcionamiento de cualquier equipo, mientras que 100W puede representar un mal funcionamiento.

La Tabla 5 muestra los estadísticos que describen a los datos real y ajustado de las cargas EQ0 y EQ2 de house 3 en la base de datos REED. Se aprecia una gran dispersión con valores extremos, pero con una buena aproximación a los datos reales.

	EQ0(real)	EQ0(prop)	EQ2(real)	EQ2(prop)
count	4000.000000	4000.000000	4000.000000	4000.000000
mean	101.642500	101.038500	4.744250	4.152750
std	45.827734	43.284627	52.098814	52.083242
min	67.000000	86.000000	0.000000	0.000000
25%	80.000000	86.000000	0.000000	0.000000
50%	92.000000	86.000000	1.000000	0.000000
75%	103.000000	86.000000	1.000000	0.000000
max	884.000000	772.000000	730.000000	727.000000

Tabla 5: Estadísticas de los datos original (real) y ajustado (prop)

De la Tabla 4, EQ0 corresponde a la carga eléctrica Tomacorriente 3 que está definido por las potencias centradas alrededor de: 86 W, 146 W, 474 W y 772 W, mientras que EQ2 Aparato electrónico 6 definido por las potencias centradas en 0 W, 106 W, 183 W, 727 W, cuya distribución es mostrada en Figura 29 donde se puede ver la gran dispersión de los datos.

Para cuantificar los errores se utiliza los estadísticos MAE (Mean Absolute Error) y RMSE (Root Mean Squared Error) aplicado a la diferencia entre las predicciones del modelo y los valores reales. La Figura 30 muestra que para EQ0 MAE puede llegar a 11W y RMSE a 16W, que representa hasta 1.4% sobre el valor máximo, de manera similar para la carga EQ2 puede llegar a 0.14% sobre el valor máximo, considerando escalas más pequeñas se encuentran errores hasta 20%. El tamaño de la ventana tiene ligera influencia sobre el error de la predicción, logrando un mejor desempeño la ventana de 90 segundos con respecto al MAE.

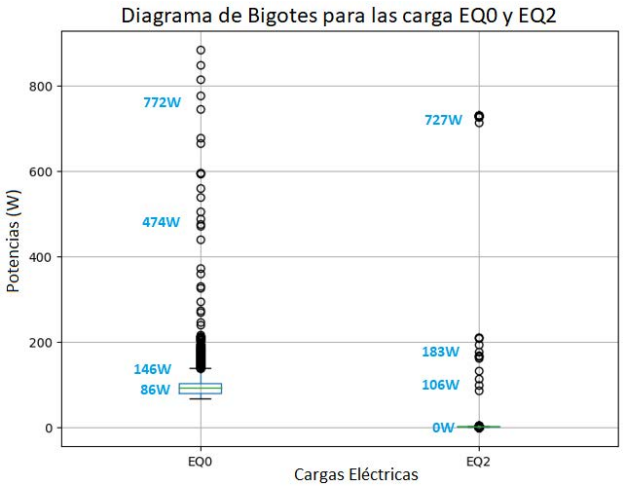


Figura 29. Diagrama de bigotes de las cargas EQ0 y EQ2

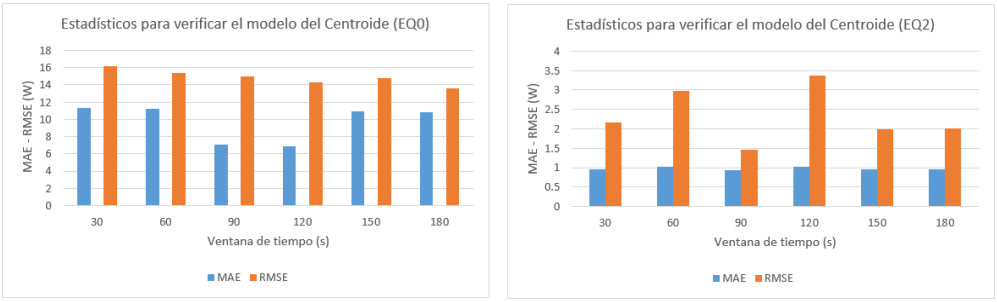


Figura 30. Cuantificación de los errores del ajuste a los centroides para las cargas EQ0 y EQ2

Sin embargo, al igual que el MAE, el RMSE no distingue entre sobreestimaciones y subestimaciones del valor objetivo, además, es sensible a los valores atípicos, ya que los errores cuadráticos pueden magnificar su influencia en la métrica general, esto evidencia la presencia de valores atípicos que resultan penalizados con un alto RMSE como se muestra en la Figura 30.

En conclusión, el comportamiento del consumo de energía de cualquier máquina eléctrica puede ser modelado con buena aproximación por valores de potencia definidos por los centroides del clúster que lo contienen en un contexto de aproximación a la energía consumida. La cantidad de clústeres está definida por la mejor representación de su consumo.

4.1.7 Obtención de las Características de la Potencia Agregada

La Figura 31 muestra la forma de la onda de la potencia agregada, que es la suma de las potencias individuales de cada aparato eléctrico. Para su caracterización, es necesario analizar y describir diversas propiedades y valores, este análisis puede llevarse a cabo en diferentes dominios, como se detalla a continuación:

- Análisis en el dominio del tiempo: Se obtiene información de amplitudes, valores máximos, mínimos y tiempos.
- Análisis en el dominio de la frecuencia: Utilizando técnicas de Fourier o Wavelets, permite identificar las frecuencias predominantes en la señal base o transformada y analizar señales con contenido no estacionarios en el tiempo.
- Análisis de la energía y la potencia: Obtenemos potencias medias y eficaces (RMS) por períodos de tiempo, también podemos extraer relaciones de señal-ruido (SNR) y calcular la energía utilizando ventanas temporales.
- Análisis estadístico: Calculamos estadísticas como la media, la desviación estándar, la mediana para describir las propiedades estadísticas de la señal.

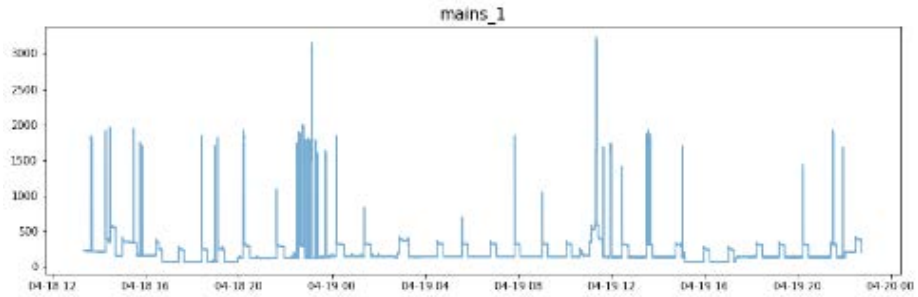


Figura 31. Grafica de la potencia total o agregada

Según se vio en 2.5 y siguiendo la propuesta de Deshmukh [60], se combinan algunas que resultan importantes para el análisis. Para la obtención de los parámetros se utiliza ventanas de tiempo según lo presentado en 4.1.5 como unidad base para la obtención de las características. Para cada ventana de tiempo y con la potencia agregada se obtiene las siguientes características: Potencia media, Desviación estándar, Hora del día, Potencia pico, Energía del período y día de la semana. Tal como se muestra en la Tabla 6.

Característica	Descripción	Unidad
μ_P	Potencia media de la señal muestreada	W
σ_P	Desviación estándar de la señal muestreada	W
t_d	Hora del día	Número
P_m	Potencia pico	W
E	Energía de la señal muestreada	W.s
D_w	Día de la semana	Número

Tabla 6. Características consideradas para la potencia agregada

Los parámetros elegidos permiten relacionar varias características, la Potencia media total (μ_P) se relaciona con el funcionamiento de los equipos en esa ventana de tiempo y su valor representa la suma de las potencias parciales, la Potencia pico o máxima (P_m) brinda información de la potencia de mayor nivel asociado a cada equipo en esa ventana de tiempo, prevalecerá la potencia del mayor consumidor, si bien estas tienen un desarrollo muy similar puede permitir detectar cambios

instantáneos asociados a condiciones transitorias o conexión de grandes consumidores como se muestra en la Figura 32.

La Energía eléctrica (E) presenta una relación entre la potencia consumida y el tamaño de la ventana de tiempo. El día de la semana (D_w) y hora del día (t_d) están relacionados con las tendencias de uso de los equipos y es posible obtener patrones de funcionamiento de las cargas eléctricas como la iluminación, el uso de máquinas lavadoras, la cocción, de forma indirecta también es posible asociar las fallas y anomalías en la instalación eléctrica.

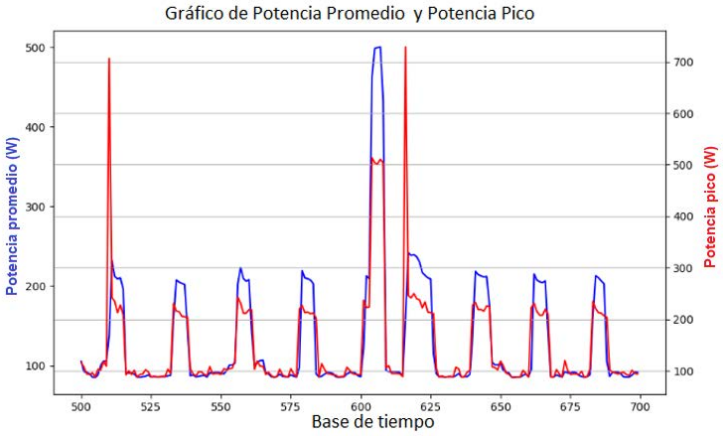


Figura 32. Relación temporal de las potencias media y pico

La Desviación Estándar (σ_P) Indica cuánto se desvían los valores individuales de la media en un conjunto de datos, el valor en cada ventana proporciona información de la dispersión, donde valores más altos indican una mayor variabilidad que puede ser causado por la conexión, desconexión o cambio de estado en un aparato eléctrico, mientras que valores más bajos indican cierta estacionalidad.

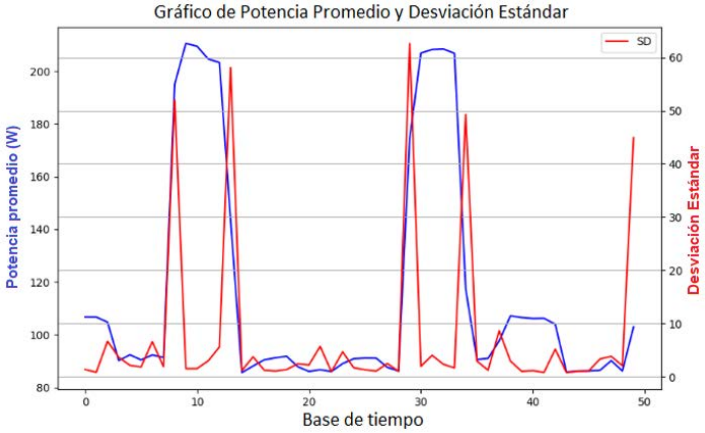


Figura 33. Relación entre la desviación estándar y la potencia promedio

Es posible hacer comparaciones de las desviaciones estándar entre diferentes ventanas para identificar patrones temporales o segmentos de tiempo con comportamientos distintos. La Figura 33 muestra cómo un cambio brusco de la potencia produce cambios en la desviación estándar.

Se pueden incluir otras características como la temperatura, el consumo de agua o la presencia de personas en la casa para mejorar la identificación del funcionamiento de los aparatos eléctricos, pero estas dependen de variables que no se pueden extraer directamente de la potencia muestreada.

4.1.8. Propuesta de Codificación Binaria Para la Potencia Agregada

El objetivo es relacionar la potencia total o agregada de la instalación eléctrica con las potencias consumidas por cada aparato eléctrico. El procedimiento propuesto es el siguiente; para cada ventana de tiempo de la potencia agregada es necesario determinar las características mostradas en la Tabla 6. Estas características están relacionadas con las potencias parciales consumidas por cada aparato eléctrico de esa instalación. El desafío se centra en representar las potencias consumidas por todos los aparatos eléctricos en una ventana de tiempo mediante un código único.

Según el análisis visto en 4.1.2 un aparato eléctrico puede ser representado mediante un número reducido y finito de estados. Utilizando los estados discretos de cada máquina para una ventana de tiempo se genera un código binario único que se relacione con la potencia agregada.

Para ilustrar la propuesta veamos los siguientes ejemplos:

Consideremos dos cargas eléctricas que trabajan simultáneamente con los niveles de potencia indicados en la Tabla 7.

	Estado 1	Estado 2	Estado 3	Estado 4
Aparato eléctrico 1	0W	20W	60W	80W
Aparato Eléctrico 2	0W	15W	50W	120W
Código binario asociado	00	01	10	11

Tabla 7. Codificación de los estados de los aparatos eléctricos.

Cada aparato eléctrico presenta 4 estados de funcionamiento por lo que usaremos una codificación de 2 bits para cada estado. Para el caso real las potencias de los estados de funcionamiento son hallados utilizando técnicas de Clustering como se vio en 4.1.4. Asignamos los siguientes códigos binarios a cada estado: Estado 1: 00, Estado 2: 01, Estado 3:10, Estado 4: 11.

Consideremos el funcionamiento simultáneo de los aparatos en una secuencia de 5 ventanas de tiempo tal como se muestra en la Figura 34

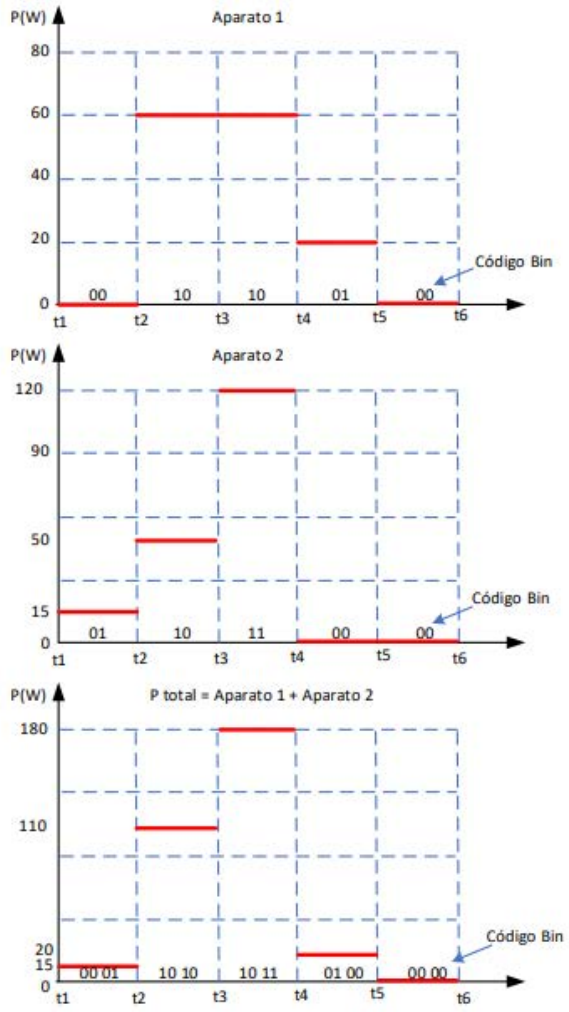


Figura 34. Funcionamiento de los aparatos en una secuencia de 5 ventanas de tiempo

La Figura 34 muestra que la potencia total (P total), que se obtiene sumando las potencias parciales de los aparatos 1 y 2. A continuación asignamos a cada ventana de tiempo un código que es una combinación de los códigos binarios parciales en esa ventana. Se indica el procedimiento para tres ventanas de tiempo:

Ventana 1 de t1 a t2:

Potencia en el aparato 1: 0W y es codificado con 00

Potencia en el aparato 2: 15W y es codificado con 01

Potencia total: 15W es codificado con 00 01

Ventana 2 de t2 a t3:

Potencia en el aparato 1: 60W y es codificado con 10

Potencia en el aparato 2: 50W y es codificado con 10

Potencia total: 110W es codificado con 10 10

Ventana 3 de t3 a t4:

Potencia en el aparato 1: 60W y es codificado con 10

Potencia en el aparato 2: 120W y es codificado con 11

Potencia total: 180W es codificado con 10 11

La codificación completa para las 5 ventas se muestra en la Tabla 8:

	Código de Carga 1	Código de Carga 2	Código de Total	Código decimal	Potencia total (W)
Ventana 1	00	01	0001	1	15
Ventana 2	10	10	1010	10	110
Ventana 3	10	11	1011	11	180
Ventana 4	01	00	0100	4	20
Ventana 5	00	00	0000	0	0

Tabla 8. Codificación binaria de la potencia total

La codificación binaria puede aplicarse de manera general a cualquier carga, para una carga de 2 estados sólo es necesario 1 bit que corresponde a los estados apagado (0) y encendido (1), en el caso de las máquinas de 4 estados o menos es necesario el uso de 2 bits ya que hay $2^2 = 4$ posibles estados. Podemos incrementar el número de bits para aumentar la cantidad de sub estados.

Consideramos un ejemplo con más detalle. La Tabla 9 considera tres aparatos eléctricos representados por 4 estados con los centroides hallados usando clustering. En un determinado instante t_x, las potencias verdaderas medidas de estos aparatos son las siguientes:

Tomacorriente 4: 121 W,

Aparato electrónico 6: 207 W,

Refrigerador 7: 610 W.

Utilizando los centroides de la Tabla 9 se puede codificar la potencia verdadera de la siguiente forma:

Tomacorriente 4 de 121 W es representado por 118 W y con código 01.

Aparato electrónico 6 de 207 W es representado por 208 W con código 10.

Refrigerador 7 de 610 W es representado por 581 W con código 10.

	Código binario			
Aparato eléctrico	00	01	10	11
Tomacorriente 4	0	118	405	1312
Aparato electrónico 6	0	729	208	131
Refrigerador 7	4	718	581	111

Tabla 9. Centroides de los aparatos eléctricos considerados en vatios (W).

Finalmente, el código único que representa la combinación de potencias de los tres aparatos eléctricos en formato binario y formato decimal es mostrado en la Tabla 10.

Tomacorriente 4	Aparato electrónico 6	Refrigerador 7	Código Binario	Código Decimal
01	10	10	011010	26

Tabla 10. Código binario y decimal de la instancia según las potencias de los aparatos eléctricos

Mediante este procedimiento se puede representar la potencia agregada en cualquier momento durante el funcionamiento del sistema eléctrico.

4.1.9. Obtención del DataFrame

Cada ventana de tiempo representa una instancia, cada instancia tiene las características extraídas a partir de la potencia medida en el circuito principal, estas son la potencia media, desviación estándar, hora del día, potencia máxima, energía absorbida y día de la semana, al mismo tiempo se ha determinado el código asociado a esa instancia, que representa los consumos individuales de cada aparato eléctrico. La Tabla 11 muestra el desarrollo.

Instanda	Características de la señal agregada						Potencia total agregada	Etiqueta	Estado de funcionamiento de las cargas					
	μP	σP	td	Pm	E	Dw			L1	L2	L3	L4	L5	L6
0	μP_0	σP_0	td_0	Pm_0	E_0	Dw_0	PT_0	E_0	$L1_{K0}$	$L2_{K0}$	$L3_{K0}$	$L4_{K0}$	$L5_{K0}$	$L6_{K0}$
1	μP_1	σP_1	td_1	Pm_1	E_1	Dw_1	PT_1	E_1	$L1_{K1}$	$L2_{K1}$	$L3_{K1}$	$L4_{K1}$	$L5_{K1}$	$L6_{K1}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n-1	μP_{n-1}	σP_{n-1}	td_{n-1}	Pm_{n-1}	E_{n-1}	Dw_{n-1}	PT_{n-1}	E_{n-1}	$L1_{Kn-1}$	$L2_{Kn-1}$	$L3_{Kn-1}$	$L4_{Kn-1}$	$L5_{Kn-1}$	$L6_{Kn-1}$
n	μP_n	σP_n	td_n	Pm_n	E_n	Dw_n	PT_n	E_n	$L1_{Kn}$	$L2_{Kn}$	$L3_{Kn}$	$L4_{Kn}$	$L5_{Kn}$	$L6_{Kn}$

Descripción

μP :Potencia media de la señal
σP :Desviación estandar de la señal
td :Hora del día
Pm :Potencia pico
E :Energía
Dw :Día de la semana

LnKi: Estado particular *i* de la carga eléctrica *n*
Es un número binario que representa uno de los valores de potencia del contenedor de potencia

Tabla 11. DataFrame para el proceso de medición no intrusiva (NILM)

4.2 Método Para la Identificación del Mejor Modelo

Obtenidas las etiquetas y características de las instancias, se sigue el procedimiento mostrado en la Figura 35 para la evaluación de los modelos y la obtención de los consumos individuales mediante el enfoque propuesto.

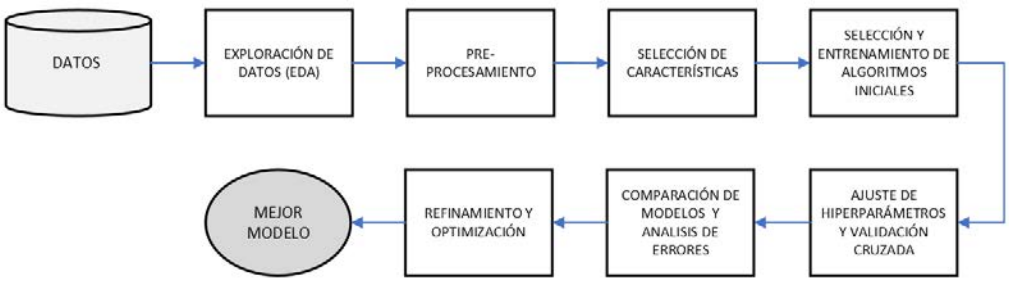


Figura 35. Procedimiento para la obtención del mejor modelo para NILM

4.2.1 Exploración de los Datos (EDA)

En 4.1.1 se ha realizado una exploración de las ondas temporales a nivel individual y agregado, haciendo énfasis en los valores estadísticos para describir el comportamiento de la potencia eléctrica, del mismo modo se ha planteado el uso de los centroides con el objetivo de minimizar las potencias instantáneas del sistema eléctrico. Mediante la aplicación de los estadísticos MAE y RMSE se ha valorado el efecto de la duración de las ventanas temporales y finalmente se ha explicado el uso de la codificación binaria en la obtención del código o etiqueta que represente las características de dicha potencia, logrando establecer el DataFrame que será utilizado como insumo para identificar los modelos adecuados para la aplicación de la medición no intrusiva NILM en el enfoque propuesto.

4.2.2 Etiquetas

Debido al funcionamiento arbitrario de cada aparato eléctrico, los centroides de potencia y en consecuencia las etiquetas representativas del DataFrame tendrán frecuencia de repetición muy variados causando un gran desbalance incluyendo datos con muy poca frecuencia de repetición. La propuesta inicial es aplicar filtros para minimizar el desbalance, en consecuencia, habrá efectos positivos y negativos que debemos analizar y evaluar:

Al aplicar un filtro para minimizar los datos de poca frecuencia de repetición se puede simplificar el modelo reduciendo su complejidad y proceso de entrenamiento, esto permitirá una fácil interpretación y una reducción del sobreajuste, en contraposición se presentará pérdida de información que puede ser relevante, inclusive afectando a las clases minoritarias dando lugar a una menor capacidad para representar correctamente la realidad.

4.2.3 Análisis de Correlación.

El análisis de correlación entre los datos de un DataFrame tiene como objetivo observar las relaciones entre las variables y su impacto en la capacidad del modelo para predecir la variable objetivo. En la Tabla 12 se presentan los resultados del análisis de correlación de las variables, donde se destacan resultados más favorables al utilizar los coeficientes de correlación de Spearman y Kendall, que son favorecidos por su robustez frente a la gran variación de escalas de valores presente en el DataFrame, al mismo tiempo se confirma la presencia de relaciones no lineales y valores atípicos identificado por un pobre resultado de la correlación de Pearson. No obstante, debemos recordar que a pesar de mostrar una correlación débil no se descarta la viabilidad de utilizar estas variables como predictores. Es importante considerar que la correlación no implica causalidad y que pueden existir otros factores más complejos que no se capturan adecuadamente en este análisis.

Matriz de Correlación de Pearson:

	pot_prom	SD	hora	pot_max	energia	dia	codigo
pot_prom	1.000000	0.327641	0.136783	0.674396	0.007969	-0.233232	0.028029
SD	0.327641	1.000000	0.049133	0.800822	-0.001266	-0.108181	0.052851
hora	0.136783	0.049133	1.000000	0.092872	0.009291	0.004109	0.077842
pot_max	0.674396	0.800822	0.092872	1.000000	0.003865	-0.185878	0.045262
energia	0.007969	-0.001266	0.009291	0.003865	1.000000	-0.014493	-0.016012
dia	-0.233232	-0.108181	0.004109	-0.185878	-0.014493	1.000000	-0.231003
codigo	0.028029	0.052851	0.077842	0.045262	-0.016012	-0.231003	1.000000

Matriz de Correlación de Spearman:

	pot_prom	SD	hora	pot_max	energia	dia	codigo
pot_prom	1.000000	0.401656	0.122138	0.957585	0.993170	-0.265222	0.171347
SD	0.401656	1.000000	0.126431	0.568770	0.398918	-0.261196	0.253944
hora	0.122138	0.126431	1.000000	0.133634	0.120998	0.011228	0.086804
pot_max	0.957585	0.568770	0.133634	1.000000	0.951677	-0.273988	0.187548
energia	0.993170	0.398918	0.120998	0.951677	1.000000	-0.263363	0.162954
dia	-0.265222	-0.261196	0.011228	-0.273988	-0.263363	1.000000	-0.228447
codigo	0.171347	0.253944	0.086804	0.187548	0.162954	-0.228447	1.000000

Matriz de Correlación de Kendall:

	pot_prom	SD	hora	pot_max	energia	dia	codigo
pot_prom	1.000000	0.276183	0.083748	0.849688	0.945583	-0.191836	0.105218
SD	0.276183	1.000000	0.090849	0.412590	0.273010	-0.192030	0.173430
hora	0.083748	0.090849	1.000000	0.091229	0.082132	0.012967	0.078594
pot_max	0.849688	0.412590	0.091229	1.000000	0.832416	-0.200319	0.111840
energia	0.945583	0.273010	0.082132	0.832416	1.000000	-0.190962	0.099101
dia	-0.191836	-0.192030	0.012967	-0.200319	-0.190962	1.000000	-0.179700
codigo	0.105218	0.173430	0.078594	0.111840	0.099101	-0.179700	1.000000

Tabla 12. Resultado de la aplicación del análisis de correlación a los datos del DataFrame

4.2.4 Datos Multiclase y Desbalanceado

La abundancia de muestras en una clase específica puede conducir a una clasificación sesgada hacia el grupo mayoritario, lo que dificulta la generalización de patrones en clases con menos muestras. El análisis de los datos revela un desequilibrio significativo entre las clases, como se muestra en la Figura 36.

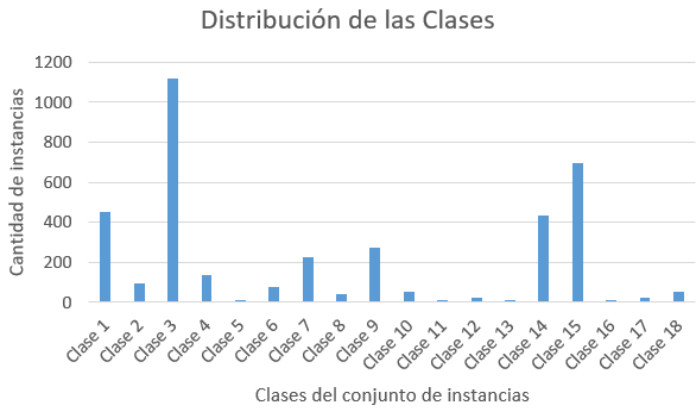


Figura 36. Cantidad de instancias por clase en el DataFrame.

Para abordar el problema del desbalance se toma como referencia a Abokadr [59], en la propuesta de utilizar la técnica de sobre muestreo para incrementar las clases minoritarias. Para la correcta aplicación se toma en consideración los siguientes aspectos:

- Utilizar validación cruzada estratificada para garantizar la correcta distribución de las clases (StratifieldKfold).
- Utilizar Pipeline para realizar procesamiento por lotes, en este caso SMOTE se aplica como uno de los pasos antes del entrenamiento en cada pliegue de la validación cruzada.

- Utilizar el parámetro class_weight en los modelos seleccionados para dar mayor importancia a las clases minoritarias.
- Disminuir la posibilidad de overfitting utilizando las curvas de validación para ajustar los hiperparámetros del modelo.
- Utilizar métricas adecuadas para conjuntos desbalanceados.
- Utilizar la Matriz de Confusión para analizar el comportamiento del modelo en cada clase.

4.2.5 Codificación de Variables Categóricas

Las variables categóricas “Día de la semana” y “Hora del día” deben ser codificadas para que puedan ser utilizadas como entrada en el algoritmo de aprendizaje automático, elegimos la codificación One-Hot Encoding para convertir cada categoría en una columna binaria que representa la presencia o ausencia de la variable, de este modo se mantiene la compatibilidad con algoritmos numéricos. Una muestra de la codificación se observa en la Tabla 13.

	Mañana	Tarde	Noche	Madru	Dom	Lun	Mar	Mie	Jue	Vie	Sab
0	0	0	1	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...
16071	1	0	0	0	0	1	0	0	0	0	0
16072	0	0	1	0	0	0	0	1	0	0	0
16073	0	1	0	0	0	1	0	0	0	0	0

[16074 rows]

Tabla 13. Codificación One-Hot Encoding

4.2.6 Elección del Algoritmo Para la Identificación del Modelo

No existe una superioridad a priori de un algoritmo de aprendizaje sobre otro, pero es posible identificar ciertas ventajas y desventajas basadas en las condiciones operativas tomadas en consideración:

- Objetivo del modelo. Realizar un proceso de clasificación para determinar la potencia eléctrica consumida por los aparatos eléctricos, mediante una codificación binaria.
- Cantidad de datos: Para el entrenamiento se ha recopilado información durante un periodo prolongado de tiempo. Durante el despliegue estos datos serán obtenidos a partir de un medidor inteligente tomando muestras a bajas frecuencias; esto limita la cantidad de muestras disponible para el reconocimiento.
- Interpretación. Es importante mantener un modelo simple y de baja complejidad que permita afinar con facilidad el modelo final.
- Recursos de hardware. El despliegue se realizará en procesadores de bajas prestaciones en forma de sistema embebido, en ese sentido los modelos simples son recomendados.

- Tipo de datos. Los datos eléctricos deben ser pre procesados para evitar señales de ruido, valores atípicos, y desbalances entre otros. La elección debe orientarse por algoritmos robustos frente a estas condiciones.
- Escalabilidad. Se debe considerar el crecimiento de la instalación eléctrica que se traduce en la operación de más aparatos eléctricos.

Tomando como base las condiciones operativas descritas realizamos una valoración inicial. En la Tabla 14, se realiza una comparación de algoritmos seleccionados.

	Algoritmo				
Característica	**Regresión**	**Random Forest**	**Máquinas de Soporte Vectorial (SVM)**	**Redes Neuronales**	**K-Nearest Neighbors (K-NN)**
Tipo de Problema	Regresión / Clasificación	Regresión / Clasificación	Regresión / Clasificación	Regresión / Clasificación	Regresión / Clasificación
Tamaño del Conjunto de Datos	Pequeño a Grande	Pequeño a Grande	Pequeño a Grande	Grande	Pequeño a Grande
Complejidad del Modelo	Baja	Moderada	Moderada/Alta	Alta	Baja
Interpretabilidad	Alta	Moderada	Moderada	Baja	Baja
Tiempo de Entrenamiento	Rápido	Moderado	Moderado	Moderado a Largo	Bajo
Manejo de Datos Faltantes y Ruido	Sensible	Robusto	Sensible	Sensible	Robusto
Escalabilidad	Alta	Moderada	Moderada	Alta	Moderada
Manejo de Datos no lineales	Sensible	Robusto	Robusto	Robusto	Robusto
Valoración	**6**	**8**	**7**	**3**	**7**

Tabla 14. Valoración de los requerimientos de los algoritmos de Aprendizaje Automático para el manejo de los datos NILM

La Tabla 15, propuesto por Rehman [61], realiza una comparación de varios algoritmos de Aprendiza Automático, en primera instancia los que están basados en probabilidades o relaciones de covarianza no son considerados para este trabajo debido a la complejidad de obtener un modelo óptimo con clases de alta correlación y relaciones no lineales, entre estos están Naive Bayes Classifier y Gaussian Process. Se toma como referencia la información de las Tablas 14 y 15 para ensayar varios algoritmos y aplicar la técnica de búsqueda en cuadrícula con el objetivo de determinar su rendimiento.

	Modelo ML	Ventajas	Desventajas
Support Vector Machine	SVM	Insensible a la dimensionalidad de los datos, buena capacidad de generalización, selección de núcleo versátil	Mayor complejidad y requisitos de memoria, dependen de los parámetros del modelo, mala interpretabilidad
Logistic Regression	LR	Modelo paramétrico, capacidad para manejar la no linealidad.	Problemas de multicolinealidad que requieren un tamaño de muestra grande
Decision tree	DT	Buena capacidad de generalización, solidez al ruido, computacionalmente más rápido, fácil de interpretar.	Problemas de sobreajuste, problemas de propagación de errores y tendencia a la dimensionalidad de los datos
Random forest	RF	Computacionalmente más rápido, robustez frente al ruido, sin ajuste de parámetros ni sobreajuste	El creciente número de árboles ralentiza el modelo.
K-nearest neighbors	k-NN	Apto para clases de modelos múltiples, simplicidad.	Confíe en el ajuste del valor k, propenso a ruido/características irrelevantes, problemas de dimensionalidad, mayores requisitos de memoria, mala interpretabilidad
Gaussian Processes	GP	Enfoque probabilístico, buen desempeño en la práctica.	Alto coste computacional
Perceptrón multicapa (Multilayer perceptron)	MLP	No paramétrico, robusto al ruido y a características irrelevantes.	Gran tiempo de entrenamiento, depende de los parámetros de entrada, difícil de interpretar
Naive Bayes Classifier	NB	Sin ajuste de parámetros, resistente a los valores faltantes, computacionalmente más rápido, requiere poca memoria	Propenso a la dimensionalidad de los datos
Linear and Quadratic Discriminant Analysis	QDA	Se calcula fácilmente, funciona bien en la práctica, sin ajuste de hiperparámetros	Largo tiempo de entrenamiento, operación compleja
Descenso de gradiente estocástico (Stochastic gradient descent)	SGD	Fácil de implementar, eficiencia, convergencia más rápida	Se requiere ajuste de hiperparámetros, sensible al escalado de funciones

Tabla 15. Ventajas y desventajas de los algoritmos de Aprendizaje Automático [61]

Los pasos para la búsqueda utilizando un proceso de Validación Cruzada es mostrado en la Figura 37.

Figura 37. Procedimiento para ensayar varios algoritmos de Aprendizaje Automático.

4.2.7 Escalamiento de Características

El análisis de las ondas temporales de potencia evidencia mucha variabilidad en los valores de sus características. Con el objetivo de no comprometer el rendimiento general de los algoritmos, se realiza el escalamiento de estas variables.

4.2.8 Búsqueda en Cuadrícula

Se utiliza la técnica de Stratified Shuffle Split para dividir los datos en conjuntos de entrenamiento y prueba de manera estratificada y aleatoria, manteniendo la proporción de clases en ambos conjuntos, seguido se realiza el proceso de pipeline con escalamiento para luego realizar el proceso de búsqueda. El procedimiento para la selección de variables con KNN se muestra en la Tabla 16.

Configuración de diccionario	Resultado de la búsqueda

KneighborsGrid = {
'kneighborsclassifier__n_neighbors': [1, 2, 5, 10],
'kneighborsclassifier__weights': ['uniform', 'distance'],
'kneighborsclassifier__algorithm': ['auto', 'ball_tree',
'kd_tree', 'brute'],
'kneighborsclassifier__p': [1, 2]
}
	'kneighborsclassifier__algorithm': 'auto',
	'kneighborsclassifier__n_neighbors': 1,
	'kneighborsclassifier__p': 1,
	'kneighborsclassifier__weights': uniform'

Tabla 16. Selección de variables para la aplicación de Grid Search CV

Más detalle sobre el desempeño de los algoritmos se logra aplicando el análisis de las curvas de validación en los casos seleccionados.

4.2.9 Curvas de Validación de los Modelos

Las Curvas de Validación permiten una exploración más detallada del rendimiento de los modelos en función de los hiperparámetros, que nos permite evitar el sobre y sub ajuste. El procedimiento seguido se muestra en la figura 38.

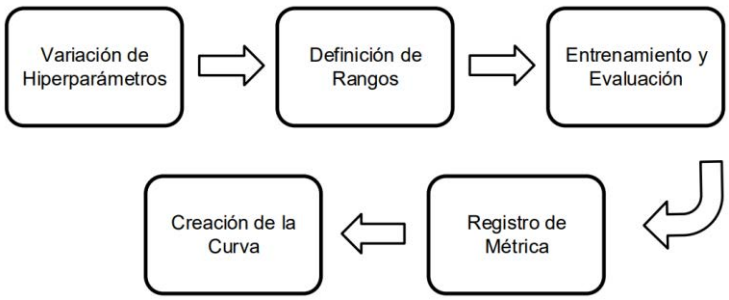


Figura 38. Procedimiento para obtener las curvas de validación

Dada la condición desbalanceada se emplea técnicas y métricas adecuadas para abordar este tipo de problemas, algunas de éstas son el uso de técnicas de regularización y manipulación de hiperparámetros para optimizar el rendimiento del modelo. Adicionalmente, se aplican métricas ponderadas que son robustas frente a condiciones de desbalance, tales como Precisión ponderada (weighted), Recall ponderado, F1-Score ponderado (macro o weighted) y el Área bajo la Curva ROC (ROC AUC).

Después de aplicar las Curvas de Validación a los hiperparámetros relevantes, se identifica aquellos que entreguen mejores resultados. Un ejemplo del análisis de estas curvas al clasificador Bagging se ilustra en la Figura 39.

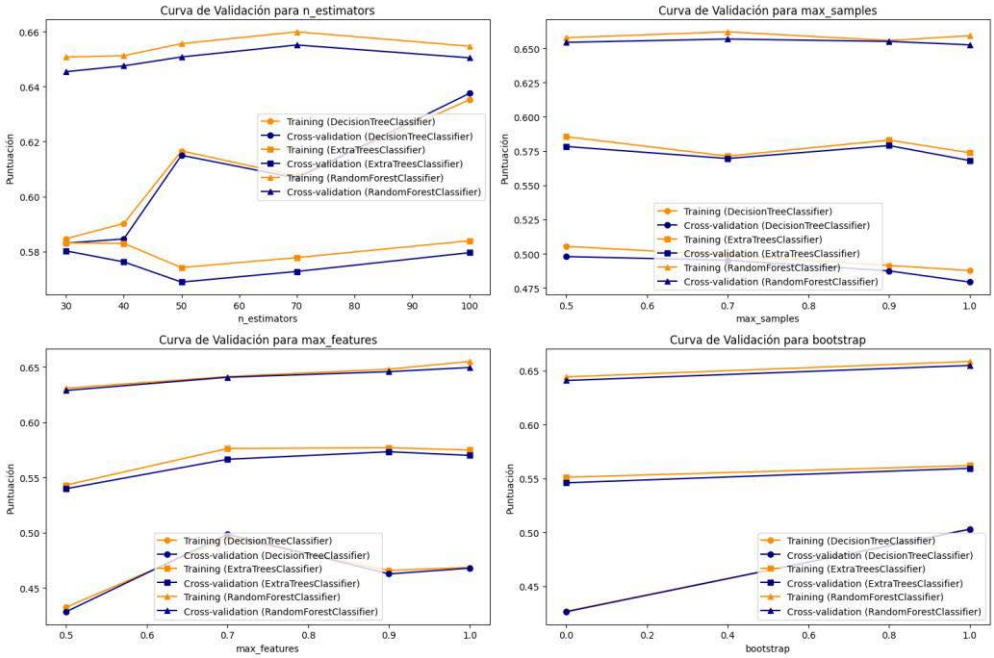


Figura 39. Curvas de validación para los parámetros seleccionados en Bagging Classifier.

4.2.10 Aplicación de los Resultados de la Curva de Validación

Una vez Identificado los mejores hiperparámetros, se calculan las medias y desviaciones estándar de los puntajes obtenidos mediante una validación cruzada. Una muestra para el clasificador RFC se presenta en la Tabla 17.

Mejores hiperparámetros	Score
Clasificador: Random Forest Classifier n_estimator:70 max_samples: 0.5 max_feature: 1 boostrap: 1	Accuracy: 0.8159 Precision: 0.8163 Recall: 0.8159 F1: 0.8138

Tabla 17. Score para Random Forest Classifier

4.2.11 Conversión de los Datos Estimados en Lecturas de Potencia Consumida

Los modelos entrenados entregan las predicciones de las potencias agregadas codificadas en formato de número entero como los mostrados en la Tabla 18. Este número debe ser decodificado para obtener la potencia individual consumida por cada aparato eléctrico.

Índice	y_predicho
0	212992
1	16384
2	196608
3	16385
....	
754	81920
755	16384
756	213376

Tabla 18. Predicciones del modelo
en formato decimal

El procedimiento para la decodificación es el siguiente: Considerando los datos predichos (y_predicho), se convierte cada código en una representación binaria de 18 bits (para 9 cargas eléctricas) dónde cada código de potencia es representado por 2 bits de la palabra binaria.

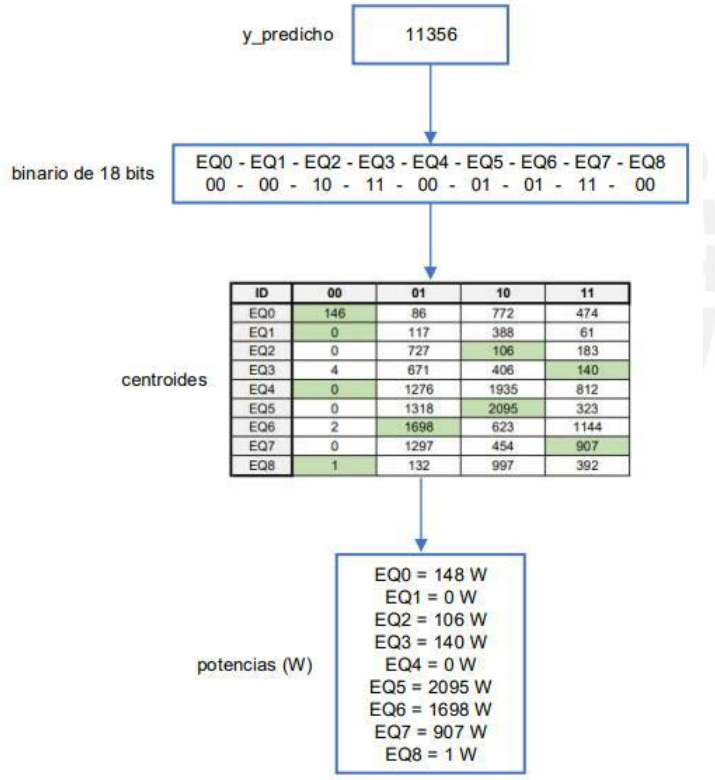


Figura 40. Procedimiento para obtener la potencia individual consumida
por las cargas eléctricas

El código binario de potencia hace correspondencia a la matriz de centroides de donde se obtiene la potencia de cada aparato eléctrico. Todo el procedimiento se ilustra en la Figura 40.

4.2.12 Métricas Utilizadas.

El desbalance de clases genera problemas en el proceso de clasificación. Entre estos está el sesgo en la predicción, que tiende a favorecer a las clases con mayor frecuencia, es decir, aquellas que representan condiciones inoperativas del aparato eléctrico. Esto causa una menor sensibilidad hacia las clases minoritarias, que suelen corresponder a estados de operación más críticos de los aparatos.

El rendimiento del modelo puede variar significativamente según la clase, y puede hacer que métricas de evaluación, como la Precisión, resulten engañosas en presencia de clases desbalanceadas. Por tanto, es importante realizar un análisis más detallado que considere métricas adecuadas y un análisis individual de cada clase.

Para mejorar la validación, se han tomado varias consideraciones. En primer lugar, se han elegido métricas de evaluación sensibles al desbalance y que incorporan los pesos de las clases, como la precisión ponderada y el F1 ponderado. En estos casos, se utiliza el parámetro class_weight para configurar estos pesos. Además, se emplea la Matriz de Confusión para obtener una visión detallada del rendimiento del modelo, que permite identificar aquellas clases que presentan mayores dificultades de clasificación.

Para asegurar una correcta verificación de la efectividad del modelo, se implementa una Validación Cruzada Estratificada. Este enfoque garantiza que cada pliegue tenga una distribución similar de clases, evitando así sesgos en la evaluación.

Adicionalmente, se puede considerar el uso de las Curvas ROC, el Área bajo la Curva (AUC) y la curva Precision-Recall como herramientas adicionales para evaluar el rendimiento del modelo en conjuntos de datos desbalanceados.

En la Tabla 19 se presenta una muestra de la aplicación de las métricas ponderadas en una predicción con Decision Tree, donde se observa que las clases con pocos miembros pueden presentar mayores problemas en la predicción.

Clase	Precision	Recall	F1-Score	Support
0	0.81	0.86	0.83	91
1	0.73	0.84	0.78	19
16384	1	0.98	0.99	224
16385	0.72	0.78	0.75	27
32768	0	0	0	2
49152	0.62	0.62	0.62	16
65536	0.79	0.91	0.85	46
65537	0.43	0.38	0.4	8
81920	0.96	0.87	0.91	55
81921	0.89	0.8	0.84	10
86017	0	0	0	2
114688	0.5	0.6	0.55	5
147456	0	0	0	2
196608	0.86	0.79	0.83	87
212992	0.96	0.95	0.95	139
212993	0	0	0	3
213184	0.6	0.75	0.67	4
245760	0.42	0.45	0.43	11

Tabla 19. Puntuación de la predicción obtenida por cada clase

CAPÍTULO V

PRUEBAS Y RESULTADOS

A continuación, se presenta la descripción práctica de los conceptos desarrollados en el capítulo anterior. El proceso inicia con una explicación detallada sobre la manipulación de datos, en la que se implementa la propuesta de extracción de características utilizando la técnica de Centroides de Potencia. Posteriormente, se construye un DataFrame que incluye las características de la variable objetivo, junto con la propuesta de codificación binaria. Este enfoque tiene como propósito identificar el modelo más eficiente bajo los lineamientos planteados. Finalmente, los resultados obtenidos son validados utilizando las métricas más adecuadas para garantizar su eficacia.

5.1 Obtención del DataFrame

La base de datos empleada en este estudio es REDD (A Public Data Set for Energy Disaggregation Research) [12]. Esta base de datos está diseñada específicamente para la tarea de desagregación de energía eléctrica y contiene información sobre el consumo total agregado y el consumo individual de circuitos derivados en varias viviendas reales, registradas durante un período de varias semanas.

Algunas características de los datos están disponibles en *http://redd.csail.mit.edu* y son:

- Datos eléctricos del tablero total registrados en alta frecuencia a 15kHz.
- Datos de hasta 24 circuitos individuales (derivados) debidamente etiquetado con el tipo de aparato o aparatos a una frecuencia de 0,5 Hz.
- Datos de hasta 20 monitores de toma corriente a una frecuencia de 1 Hz orientados al registro de aparatos electrónicos donde varios dispositivos se agrupan en un solo circuito.

La Tabla 20 muestra los equipos instalados en house_3 y la Figura 41 muestra los datos obtenidos en el lapso de 24 horas.

House	Monitors	Device Categories
1	20	Electronics, Lighting, Refrigerator, Disposal, Dishwasher, Furnace, Washer Dryer, Smoke Alarms, Bathroom GFI, Kitchen Outlets, Microwave
2	19	Lighting, Refrigerator, Dishwasher, Washer Dryer, Bathroom GFI, Kitchen Outlets, Oven, Microwave, Electric Heat, Stove
3	24	Electronics, Lighting, Refrigerator, Disposal, Dishwasher, Furnace, Washer Dryer, Bathroom GFI, Kitchen Outlets, Microwave, Electric Heat, Outdoor Outlets
4	19	Lighting, Dishwasher, Furnace, Washer Dryer, Smoke Alarms, Bathroom GFI, Kitchen Outlets, Stove, Disposal, Air Conditioning
5	10	Lighting, Refrigerator, Disposal, Dishwasher, Washer Dryer, Kitchen Outlets, Microwave, Stove

Tabla 20. Aparatos y dispositivos eléctricos de las casas de la base de datos REDD [12]

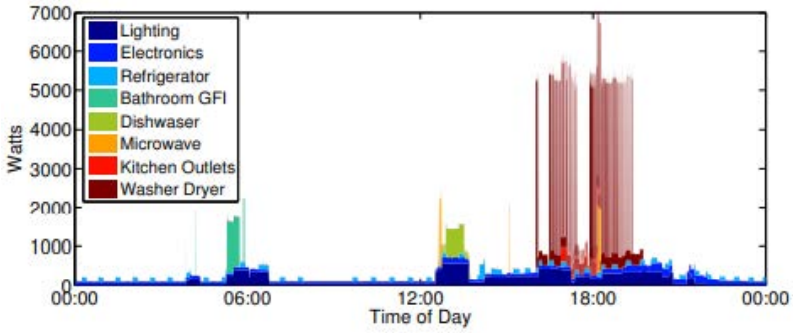


Figura 41. REDD. Consumo de energía eléctrica en un período de 24 horas [12]

REDD contiene dos tipos principales de datos de electricidad: tensión/corriente de alta frecuencia y datos de potencia de baja frecuencia, todo agrupado en tres directorios:
/low_freq/: Datos de potencia muestreados en baja frecuencia.
/high_freq/: Datos de corriente y voltaje muestreada en alta frecuencia.
/high_freq_raw/: Formas de onda de corriente y voltaje sin procesar.
Se utilizaron los datos de potencia registrado a una frecuencia de una vez por segundo para circuitos individuales y una vez cada tres segundos para los circuitos principales, es decir con muestreos de baja frecuencia y considerando el funcionamiento de estado estable de los aparatos eléctricos. El directorio principal consta de varios sub directorios *house_i*, cada uno de los cuales contiene todas las lecturas para una sola casa. Cada sub directorio *house_i*, consta de un archivo *labels.dat* y varios archivos *channel_i.dat* correspondiente a los datos del aparato eléctrico como el refrigerador, la iluminación entre otros tal como se muestra en la Figura 42.

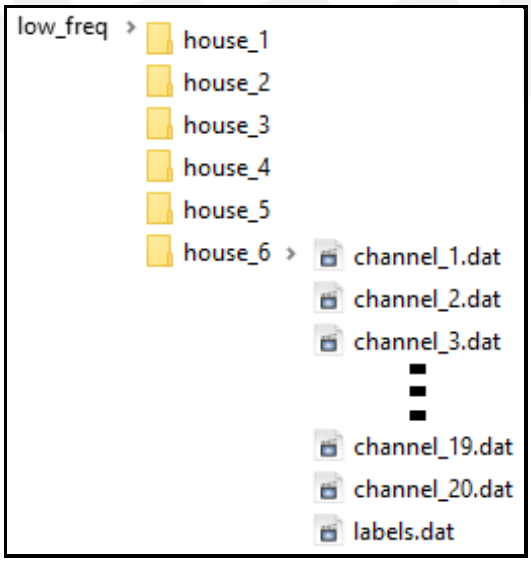


Figura 42. Estructura del directorio de los datos REDD y marcas de tiempo UTC

Se trabajó con 9 de las 24 cargas eléctricas de house 3, según se muestra en la Tabla 21.

ID	Aparato eléctrico
EQ0	Tomacorriente 3
EQ1	Tomacorriente 4
EQ2	Aparato electrónico 6
EQ3	Refrigerador 7
EQ4	Horno 10
EQ5	Iluminación 11
EQ6	Lavadora 13
EQ7	Iluminación 17
EQ8	Iluminación 19 (4 cluster)
EQ8	Iluminación 19 (5 cluster)

Tabla 21. Cargas eléctricas seleccionadas de REDD

Los datos de potencia P(W) y tiempo UTC (Universal Time Coordinated) se presentan en formato de texto como se muestra en la Figura 43. Son 360000 datos que corresponden a muestras desde sábado 16 de abril de 2011 hasta el lunes 30 de mayo de 2011 con muestras cada 3 segundos en los circuitos individuales y cada segundo en el circuito principal.

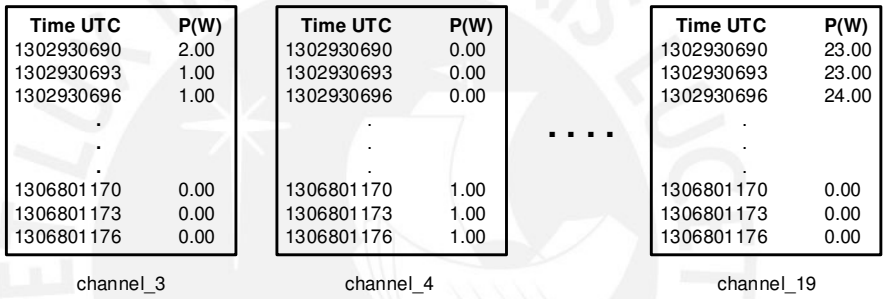


Figura 43. Archivos de texto que contiene los datos de los aparatos eléctricos

Se trabajó con 120000 datos comprendidos entre el sábado 23 de abril de 2011 al miércoles 18 de mayo de 2011, estas fueron organizadas en tres matrices: (a) Tiempos, (b) Potencia individual y (c) potencia agregada o total, como se muestra en la Figura 44.

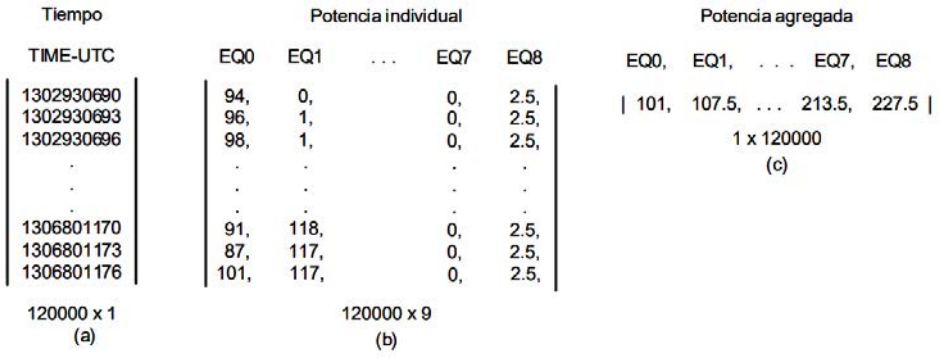


Figura 44. a) Matriz de tiempos, b) Matriz de potencias individuales, c) Matriz de potencia agregada

De acuerdo al análisis en 4.1.6 se agruparon los datos por ventanas de tiempo de 90 segundos, en consecuencia, cada instancia contendrá 30 datos de potencia para cada una de las 9 cargas eléctricas (EQ0 a EQ8) como se muestra en la Figura 45.

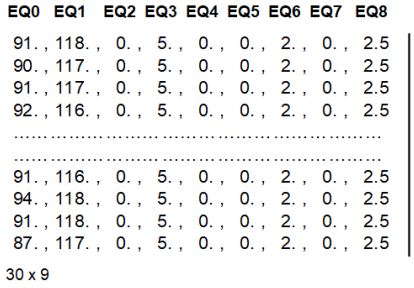


Figura 45. Matriz para formar la instancia

Cada ventana de tiempo se considera una instancia, que constituye la base para el procesamiento de los datos. A partir de estas ventanas, se calcularon las potencias promedio de cada carga, lo que permitió determinar los centroides representativos de la instalación eléctrica. La Figura 46 ilustra en detalle el objetivo de este análisis.

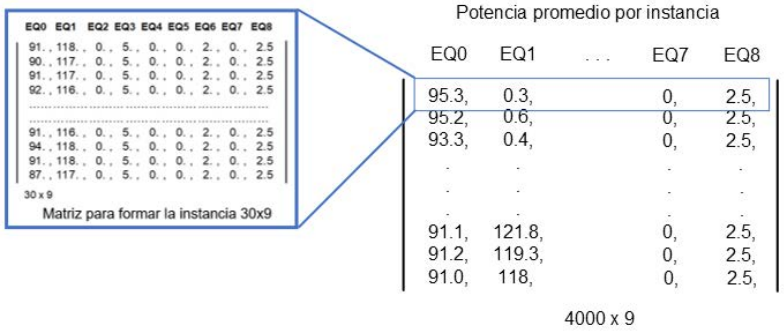


Figura 46. Obtención de las potencias promedio para cada carga

Se aplicó la técnica del Codo junto con los datos de potencias promedio para identificar los centroides de potencia que caracterizan el comportamiento de cada carga eléctrica. La definición de esta técnica y su relación con la onda temporal de potencia se abordaron en la Sección 4.1.2.

El diagrama de bloques presentado en la Figura 47 describe el procedimiento para determinar los centroides de los clústeres que representan los subestados de la carga eléctrica.

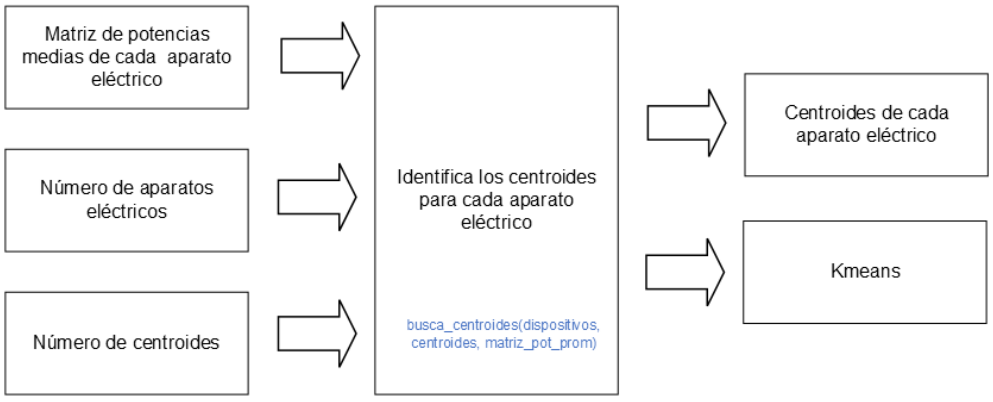


Figura 47. Procedimiento para obtener los clusters de potencia

La aplicación de la función KMeans y los resultados obtenidos se muestra en Figura 48

```
# Aplicación de k-means con k = 4.
kmeans_total[eq] = KMeans(n_clusters=centroides, n_init=10).fit(equipo)
un_centroide = kmeans_total[eq].cluster_centers_
valores_centroides[eq]=un_centroide[:,1]
```

(a)

```
      C1    C2    C3    C4
EQ0 [[ 79,  153,  627,  101]
EQ1 [ 117,    1,  388,   61]
EQ2 [  0,  728,  184,  107]
EQ3 [  5,  672,  140,  406]
EQ4 [  0,  970, 2081, 1530]
EQ5 [  0, 2096, 1319,  323]
EQ6 [  2, 1144,  623, 1698]
EQ7 [  0, 1297,  389,  868]
EQ8 [  2,  133,  997,  392]]
```

(b)

Figura 48. Obtención de los clústeres de potencia

La potencia total es agrupada en ventanas de 90 segundos, con 30 muestras por ventana y cada ventana corresponde a una instancia como se muestra en la Figura 49.

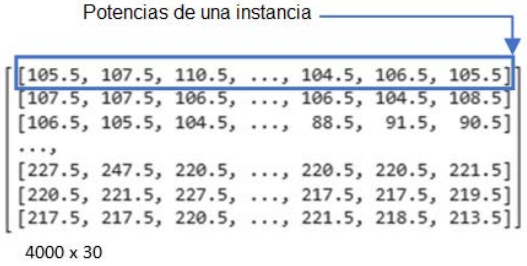


Figura 49. Potencias agregadas de una instancia

Una vez identificadas las matrices de Potencia Total y Potencia Individual por instancia, como se muestra en la Figura 50, se procede a construir la matriz de etiquetas y características que será utilizada durante el proceso de entrenamiento.

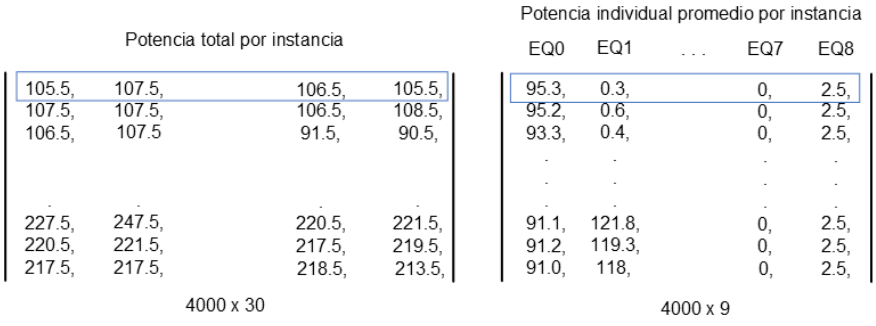


Figura 50. Matrices de potencias

El procedimiento mostrado en el diagrama de bloques de la Figura 51 se utilizó para obtener las etiquetas y las potencias de las cargas eléctricas ajustadas al centroide en cada instancia de tiempo. Las etiquetas se generaron en formatos binario y decimal.

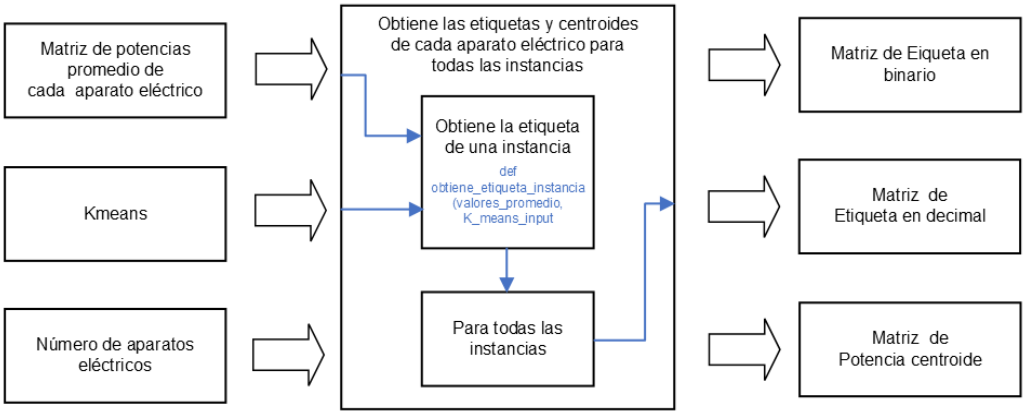


Figura 51. Obtención de las etiquetas y centroides

La Figura 52 muestra un resultado parcial en el que se observa la aproximación de las potencias centroides a las potencias promedio reales. Por ejemplo, para la carga EQ0, la potencia promedio de 95.3 W se ajusta al valor del centroide de 101 W. Asimismo, para la carga EQ1, las potencias promedio de 119.3 W, 121.8 W y 118 W se ajustan a un único valor de 117 W, correspondiente al centroide más cercano a dichas potencias.

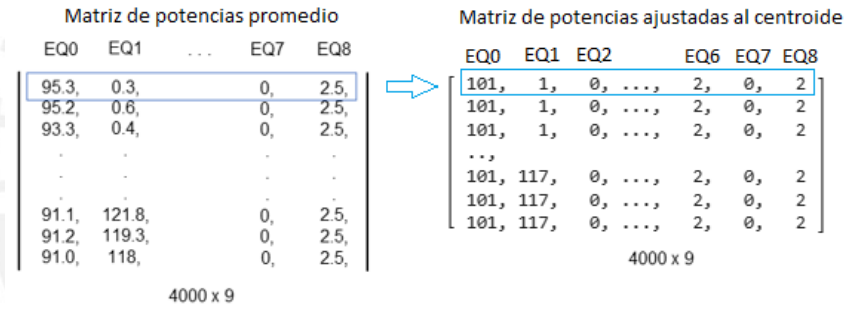


Figura 52. Matriz de potencias ajustadas al centroide

Al considerar las potencias ajustadas a los centroides, se logró reducir los estados de la instalación eléctrica. Aplicando el método propuesto en la sección 4.1.8, se llevó a cabo la codificación de los estados para cada instancia de los datos, lo que permitió obtener las etiquetas en formatos binario y decimal. La Figura 53 presenta un conjunto de etiquetas, donde la etiqueta $212992_{(10)}$ se representa en formato binario como $110100000000000000_{(2)}$.

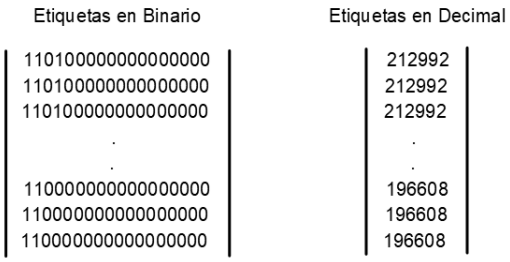


Figura 53. Matriz de etiquetas en formato binario y su representación decimal

Finalmente, la Figura 54 muestra la secuencia completa para obtener las potencias centroides y las etiquetas.

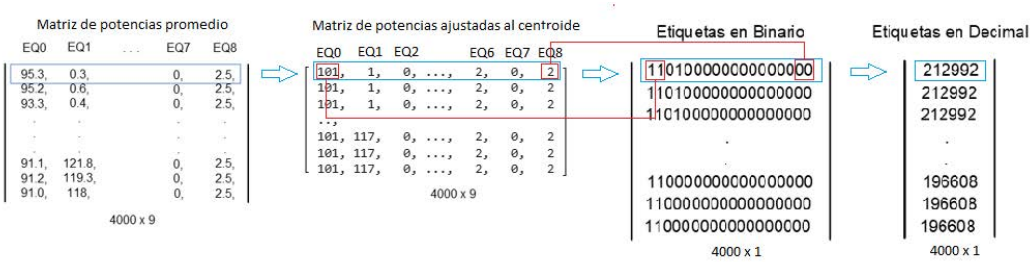


Figura 54. Proceso de obtención de las etiquetas

5.2 Obtención de las Características de la Potencia Total o Agregada

La potencia total o agregada es la información disponible para el proceso NILM. Para cada instancia de la potencia agregada, y utilizando las marcas de tiempo, se genera la matriz de características siguiendo el procedimiento ilustrado en la Figura 55.

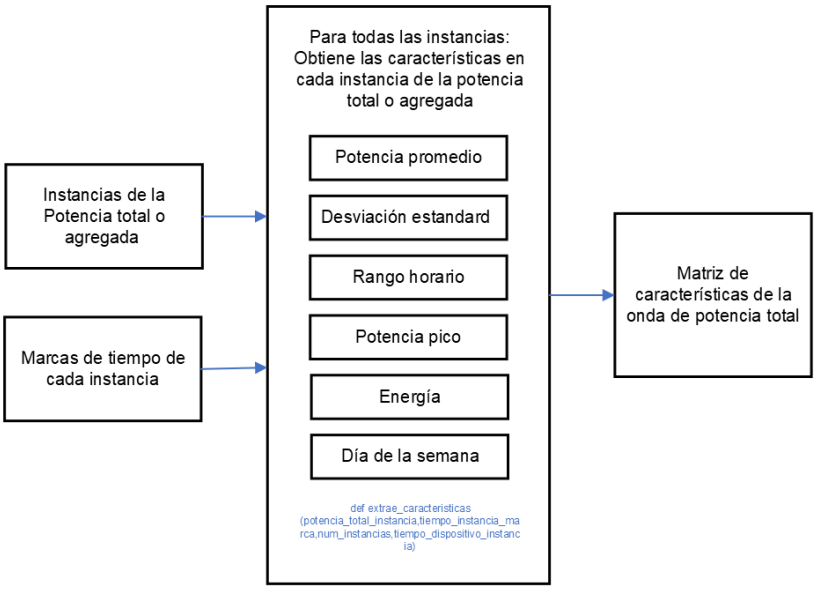


Figura 55. Proceso para la obtención de las características de la instancia

La aplicación de este procedimiento genera los valores presentados en la Tabla 22. En ella se puede visualizar la matriz que servirá como base para el entrenamiento y la validación.

	Pot_media (μ_P)	SD (σ_P)	Hora (t_d)	Pot_pico (P_m)	Energía (E)	Día (D_w)	Código (Etiqueta)
0	106.7	1.3759845	1	110.5	11755.5	5	196608
1	106.666667	0.85958646	1	108.5	11838	5	196608
2	104.766667	6.6929482	1	116.5	11570	5	196608
3	90.15	3.65410728	1	99.5	9967.25	5	0
. . .							
3997	224.5	16.7729942	4	308.5	24837.5	2	212992
3998	220.066667	2.06047459	4	227.5	23990	2	212992
3999	218.533333	3.53537677	4	234.5	24046.5	2	212992

4000 x 7

Tabla 22. DataFrame para entrenar e modelo

5.3 Pre Procesamiento

5.3.1 Filtrado.

Una revisión de los datos reveló que estos están compuestos por 4000 registros, los cuales se pueden agrupar en 109 categorías distintas. Con el objetivo de simplificar el modelo y mantener una capacidad predictiva óptima sobre los nuevos datos, se analizó la relación entre los distintos grupos y su frecuencia de aparición. Esta relación se muestra en la gráfica de tendencia de la Figura 56, donde se observa una variación mínima en los grupos con una frecuencia de repetición inferior a 10. Basándose en esta observación, se implementó un filtro para eliminar los grupos con menos de 10 repeticiones, reduciendo así el número de categorías de 109 a 18, como se muestra en la Tabla 23. Esto resultó en 3750 instancias disponibles para el entrenamiento del modelo.

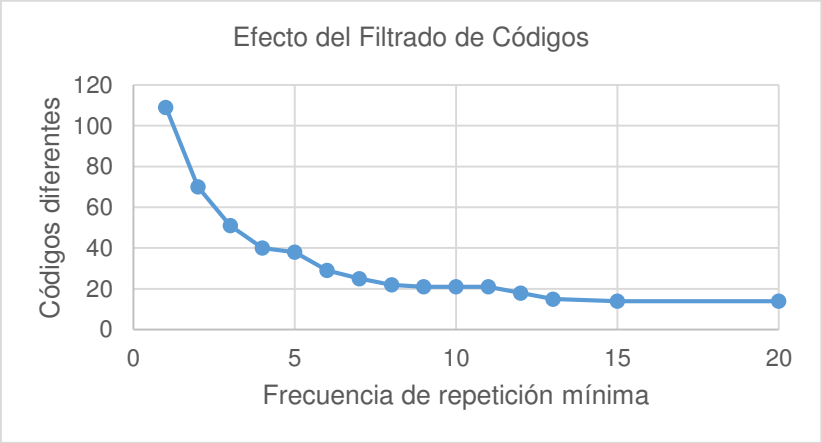


Figura 56. Gráfica de Tendencia de grupos diferentes

Código	Frecuencia
0	453
1	95
16384	1117
16385	136
32768	12
49152	79

Código	Frecuencia
65536	227
65537	42
81920	273
81921	52
86017	12
114688	24

Código	Frecuencia
147456	11
196608	434
212992	695
212993	13
213184	22
245760	55

Tabla 23. Grupos y frecuencia de repetición de los datos del DataFrame

5.3.2 Partición de los Datos.

Se realizó la partición de los datos para entrenamiento y pruebas utilizando la función Train-test Split de Scikit-Learn con el objetivo de obtener una evaluación objetiva del rendimiento del modelo, disminuyendo la posibilidad de sobreajuste y garantizar una

representación más precisa de la distribución de las clases. Luego de la aplicación se han formado 3000 muestras para entrenamiento y 750 para validación.

5.3.3 Aplicación de SMOTE

La Tabla 22 muestra un desbalance significativo entre las clases, lo que podría llevar a que el modelo se enfoque principalmente en las clases mayoritarias, descuidando las clases minoritarias. Para abordar este problema, se aplicó la técnica de SMOTE para generar datos sintéticos y equilibrar las clases durante el entrenamiento, reduciendo así el riesgo de sobreajuste.

El procedimiento consiste en aplicar SMOTE para generar instancias sintéticas de las clases minoritarias en cada pliegue de entrenamiento como se muestra en la Figura 57. Esto garantiza que cada pliegue mantenga una distribución balanceada de clases. Posteriormente, el modelo se validó utilizando métricas apropiadas en el pliegue correspondiente sin aplicar SMOTE, para evaluar su rendimiento en un conjunto de datos no balanceado.

```
pipeline = Pipeline([
    ('smote', SMOTE(k_neighbors=n_neighbors)), # Aplicar SMOTE
    ('scaler', StandardScaler()),  # Estandarizar los datos
    ('classifier', classifier)  # Clasificador dinámico ])
```

Figura 57. Pipeline incluyendo SMOTE para evaluar cada pliegue

5.3.4 Datos Categóricos

Se ha codificado tanto el día de la semana como el rango de horas de funcionamiento de los aparatos eléctricos utilizando números consecutivos, tal como se muestra en la Tabla 24.

Variable categórica	Mañana	Tarde	Noche	Madrugada
Hora del día (Horas)	06:00 – 12:00	12:00 -18:00	18:00 – 24:00	24:00 – 06:00
Código asignado	1	2	3	4

Variable categórica	Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
Día de la semana (Localtime)	0	1	2	3	4	5	6

Tabla 24. Representación de período del día (hora), y día de la semana (día)

Las variables categóricas fueron transformadas mediante el método "one-hot encoding", como se observa en la Figura 58. Esta técnica garantiza una representación adecuada y equitativa con el objetivo de evitar posibles sesgos

```
hora_array = dtrain['hora'].values
name_hora = ['Mañana', 'Tarde', 'Noche', 'Madru']  ## array([1, 2, 3, 4])
label_binarizer = sklearn.preprocessing.LabelBinarizer()
label_binarizer.fit(range(1,max(hora_array)+1))
cod_hora = label_binarizer.transform(hora_array)
cod_hora _df = pd.DataFrame(cod_hora, columns=name_hora)
```

	Mañana	Tarde	Noche	Madrugada
0	1	0	0	0
1	1	0	0	0
2	1	0	0	0
23454	1	0	0	0
23455	0	1	0	0
23456	0	0	1	0

Figura 58. Manejo de los datos categóricos

Finalmente, el DataFrame para el entrenamiento es mostrado en la Tabla 24.

Id	pot_prom	SD	pot_max	energia	Mañana	Tarde	Noche	Madru	Dom	Lun	Mar	Mie	Jue	Vie	Sab	codigo
0	106.7	1.375984	110.5	11755.5	1	0	0	0	0	0	0	0	0	1	0	212992
1	106.666667	0.859586	108.5	11838	1	0	0	0	0	0	0	0	0	1	0	212992
2	104.766667	6.692948	116.5	11570	1	0	0	0	0	0	0	0	0	1	0	212992
3	90.15	3.654107	99.5	9967.25	1	0	0	0	0	0	0	0	0	1	0	16384
4	92.366667	2.156128	100.5	10257	1	0	0	0	0	0	0	0	0	1	0	16384
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3745	211.2	48.598457	247.5	22888.75	0	0	0	1	0	0	1	0	0	0	0	196608
3746	222.95	5.167769	233.5	24533.25	0	0	0	1	0	0	1	0	0	0	0	196608
3747	224.5	16.772994	308.5	24837.5	0	0	0	1	0	0	1	0	0	0	0	196608
3748	220.066667	2.060475	227.5	23990	0	0	0	1	0	0	1	0	0	0	0	196608
3749	218.533333	3.535377	234.5	24046.5	0	0	0	1	0	0	1	0	0	0	0	196608

Tabla 25. DataFrame con pre procesamiento para realizar el entrenamiento

5.4 Aplicación de GridSearch

Se seleccionaron varios algoritmos de aprendizaje como punto de partida para evaluar y determinar cuál de ellos es el más eficiente bajo el enfoque propuesto. El objetivo principal era obtener una comprensión clara y completa del rendimiento del modelo durante el proceso de clasificación. Para lograr esto, fue fundamental encontrar la mejor combinación de hiperparámetros, por lo que se utilizó la técnica de GridSearch con la métrica balanced_accuracy. Se definió una amplia combinación de valores en una búsqueda automatizada, empleando la técnica de validación cruzada para mejorar la precisión. De este modo, se minimizó la posibilidad de sobreajuste al evaluar el rendimiento en diferentes subconjuntos. Parte de esta función y los resultados obtenidos se muestran en la Figura 59 y Tabla 26, respectivamente.

```
def grid_search(classifier, parameters):
    clf = classifier
    grid = GridSearchCV(clf, param_grid=parameters,
    scoring='balanced_accuracy', cv=5, n_jobs=-1, verbose=2)
    grid.fit(X_train, y_train)
```

Figura 59. Función GridSearchCV

Algoritmo	Mejores Parámetros	
Logistic Regression (LR)	C: 2 weight: balanced max_iter: 200	multi_class: multinomial solver: lbfgs
Decision Tree Classifier (DTC)	criterion: gini max_depth: None max_features: None min_samples_leaf: 1	min_samples_split: 2 min_weight_fraction_leaf: 0.0 random_state: 1 splitter: best
KNeighbors Classifier (KNC)	algorithm: auto n_neighbors: 1	p: 1 weights: uniform
Random Forest Classifier (RFC)	bootstrap: False criterion: gini max_depth: None	max_features: 5 min_samples_leaf: 1 min_samples_split: 3
Ada Boost Classifier (ABC)	algorithm: SAMME base_estimator: RandomForestClassifier() learning_rate: 3 n_estimators': 30, random_state: None	
Support Vector Machines (SVC)	C: 10 class_weight: balanced decision_function_shape: ovr	kernel: linear _tol: 0.01
Gradient Boosting Classifier (GBC)	learning_rate: 0.1 loss: deviance max_depth: 5 max_features: log2	min_samples_leaf: 1 min_samples_split: 5 n_estimators: 100 subsample: 0.8
Stochastic Gradient Descent Classifier (SGD)	alpha: 0.001 eta0: 0.5, learning_rate: optimal	loss: log max_iter: 200 n_iter_no_change: 10
Perceptron (PCP)	alpha: 0.001 fit_intercept: True penalty: None	random_state: 1, shuffle: False, warm_start: True
Passive Aggressive Classifier (PAC)	C: 12915.496650148827 early_stopping: False fit_intercept: True loss: squared_hinge	max_iter: 200 shuffle: True tol: 0.1 validation_fraction: 0.1
Bagging Classifier (BC)	bootstrap: False bootstrap_features: False max_features: 0.75	max_samples: 5000 random_state: 1
Extra Tree Classifier (ETC)	criterion: entropy max_depth: None max_features: auto	min_samples_leaf: 1 min_samples_split: 2 n_estimators: 200

Tabla 26. Resultados de la aplicación de GridSearchCV

La Figura 60 muestra las puntuaciones que obtuvieron los algoritmos seleccionados luego de la ejecución de GridSearchCV

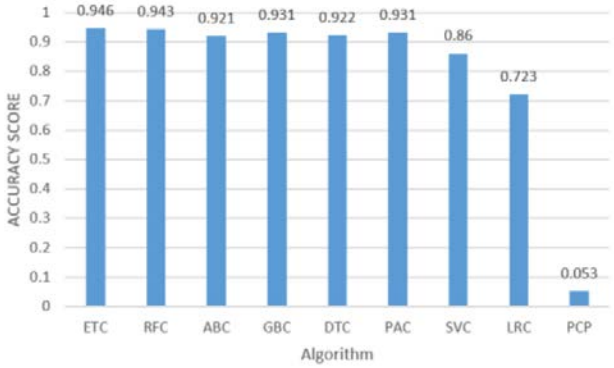


Figura 60. Puntuación obtenida mediante GridSeachCV

Se observa que los métodos de ensamble basados en árboles, como Extra Trees Classifier (ETC), Random Forest Classifier (RFC) y Gradient Boosting Classifier (GBC), obtienen un mejor desempeño, destacándose con puntuaciones más altas para este conjunto de datos específico. Estos modelos aprovechan la ventaja de combinar las predicciones de los clasificadores base, lo cual parece ser particularmente efectivo en este contexto. Asimismo, otros modelos también muestran un buen rendimiento, como Decision Tree Classifier (DTC), Ada Boost Classifier (ABC) y Passive Aggressive Classifier (PAC), que alcanzaron

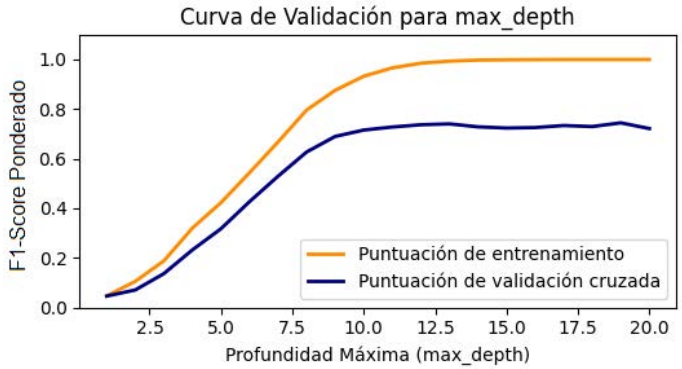
puntuaciones alrededor de 0.92. En el caso de ABC y PAC, comparten la característica de mejorar iterativamente el modelo de clasificación, adaptándose a problemas de clasificación desbalanceada. Por otro lado, los modelos que mostraron un rendimiento menos efectivo en este primer acercamiento fueron Support Vector Classifier (SVC), que podría tener un rendimiento bajo debido al desbalance de clases y su sensibilidad a la distribución desigual de los datos. También, Logistic Regression (LR) y Perceptron Classifier (PCP) obtuvieron puntuaciones más bajas, lo que sugiere que son menos adecuados para este conjunto de datos posiblemente debido a su enfoque limitado para ajustar los pesos y su falta de capacidad para priorizar las clases menos representadas.

En cuanto a los tiempos de procesamiento, algoritmos como Extra Trees Classifier (ETC) y K-Neighbors Classifier (KNC) presentan tiempos significativamente más altos, con mayor costo computacional y en consecuencia requieren mayores recursos de hardware. Por otro lado, algoritmos como Random Forest Classifier (RFC), Decision Tree Classifier (DTC) y Ada Boost Classifier (ABC) no requieren tantos recursos computacionales y resultan ser más eficientes en términos de tiempo de procesamiento, haciendo que su uso sea más práctico en contextos donde la eficiencia computacional es una prioridad.

5.5 Aplicación de las Curvas de Validación

El uso de GridSearch permitió obtener una primera aproximación general al desempeño de los modelos. Posteriormente, se llevó a cabo una evaluación más detallada, que incluyó aspectos como la interpretabilidad de los modelos y su sensibilidad a hiperparámetros específicos. Basándonos en los resultados obtenidos previamente, se seleccionaron los algoritmos con mejor desempeño para aplicar el método de Curvas de Validación, con el objetivo de identificar los valores óptimos de los hiperparámetros y mejorar el rendimiento del modelo.

Parte del procedimiento y la respuesta para *Decisión Tree Classifier* y el hiperparámetro *max_depth* se muestra en la Figura 61.



61 a) Curva de Validación para max_depth en Decisión Tree Classifier

```
train_scores_max_depth, test_scores_max_depth = validation_curve(dt_classifier, X, y,
param_name = "decisiontreeclassifier__max_depth", param_range =
param_range_max_depth, cv=5, scoring="f1_weighted", n_jobs=-1)
```

Figura 61 b) Curva de validación, extracto del código aplicado

A continuación, se muestra un resumen de los resultados.

Algoritmo	Hiperparámetro	
Decisión Tree	criterion:	Entropy
	max_depth:	8
	max_features:	5
	min_samples_leaf:	3
	min_samples_split:	5
	min_weight_fraction_leaf:	0
	random_state:	1
	splitter:	best
KNeighbors Classifier	algorithm:	auto
	n_neighbors:	3
	p:	1
	weights:	distance
Random Forest Classifier	max_depth:	8
	min_samples_split:	3
	min_samples_leaf:	3
	max_features:	4
	bootstrap:	False
	criterion:	Entropy
Gradient Boosting Classifier	max_depth:	10
	min_samples_split:	2
	min_samples_leaf:	2
	max_features:	5
	criterion:	friedman_mse
	n_estimators:	100
Support Vector Classifier	C:	20
	kernel:	linear
	gamma:	10
	degree:	1
Extra Trees Classifier	max_depth:	10
	min_samples_split:	3
	min_samples_leaf:	2
	max_features:	4
	criterion:	entropy
	n_estimators:	50

Tabla 27. Resultados de la aplicación de las Curvas de Validación

Tomando como referencia los resultados obtenidos de las curvas de validación presentadas en la Tabla 27, se utilizaron los hiperparámetros identificados para realizar un procedimiento de validación cruzada, con el objetivo de estimar con mayor precisión las puntuaciones de cada modelo seleccionado. Las métricas evaluadas incluyeron F1-score, recall, precisión y exactitud (accuracy). Los resultados, ilustrados en la Figura 62, evidencian una clara superioridad del modelo Random Forest Classifier (RFC) y de Gradient Boosting Classifier (GBC) en comparación con los otros.

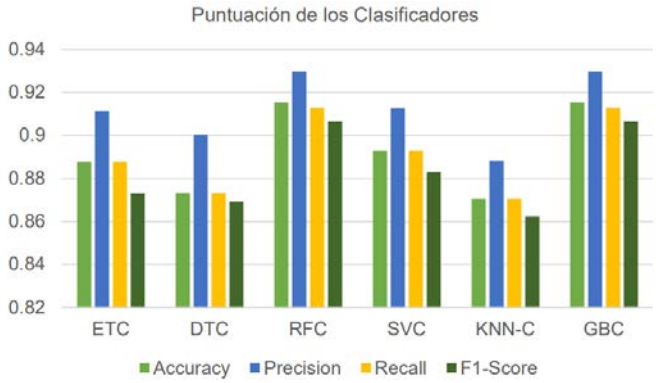


Figura 62. Puntuación obtenida mediante validación cruzada

5.6 Considerando Datos de Validación

Para seleccionar el modelo más adecuado para el problema planteado, se evaluó su rendimiento utilizando datos de validación no vistos durante el entrenamiento. Esto permitió medir la capacidad de generalización de los modelos, evitar el sobreajuste y realizar una comparación justa entre los modelos finalistas.

El procedimiento inició con el pre procesamiento de los datos, que incluyó tareas como el filtrado, la codificación de variables categóricas y el escalado de las características hasta obtener el DataFrame mostrado en la Figura 63. Dado que el objetivo era evaluar la efectividad del modelo en condiciones reales, no se aplicó sobre muestreo, trabajando directamente con un conjunto de datos desbalanceados. Posteriormente, se generó la matriz de confusión para cada uno de los modelos presentados en la Tabla 26, lo que permitió realizar un análisis detallado de cada clase.

	pot_prom	SD	pot_max	energia	Mañana	Tarde	Noche	Madru	Dom	Lun	Mar	Mie	Jue	Vie	Sab	codigo
0	171.066667	4.751725	180.5	19142.50	0	0	1	0	1	0	0	0	0	0	0	81920
1	472.783333	181.434868	663.5	50677.25	0	0	0	1	0	0	0	1	0	0	0	65537
2	195.300000	28.086058	241.5	21840.50	0	0	1	0	1	0	0	0	0	0	0	81920
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
754	162.633333	66.632691	232.5	18510.00	0	0	1	0	0	0	0	0	0	0	1	49152
755	152.300000	2.386071	157.5	15989.50	1	0	0	0	0	0	0	1	0	0	0	81920
756	158.300000	56.845639	216.5	17371.50	0	1	0	0	1	0	0	0	0	0	0	49152

757 rows × 16 columns

Figura 63. DataFrame de validación

Las métricas seleccionadas se indican en la Figura 64:

```
metrics = {
    'accuracy': make_scorer(accuracy_score),
    'precision': make_scorer(precision_score, average='weighted', zero_division=1),
    'recall': make_scorer(recall_score, average='weighted', zero_division=1),
    'f1': make_scorer(f1_score, average='weighted')   }
```

Figura 64. Métricas para la evaluación de los modelos

Estas métricas fueron seleccionadas debido a su idoneidad para manejar datos de tipo multi clase y desbalanceados. Es importante considerar que Precisión puede ser engañosa en este tipo de conjuntos, ya que puede ser elevada incluso si el modelo clasifica incorrectamente muchas instancias de la clase minoritaria. Recall mide la capacidad del modelo para identificar todas las instancias de la clase positiva, pero no es suficiente por sí solo para evaluar el rendimiento en un conjunto de datos desbalanceado. Un alto Recall puede lograrse prediciendo todas las instancias como pertenecientes a la clase mayoritaria, lo cual no sería útil en situaciones desbalanceadas. F1-score, en cambio, combina tanto Precisión como Recall, proporcionando así una medida equilibrada del rendimiento del modelo. Esta métrica es particularmente útil en conjuntos desbalanceados porque considera tanto los falsos positivos como los falsos negativos, penalizando fuertemente las discrepancias entre estas dos métricas.

5.7 Análisis de la Matriz de Confusión

Observamos el desempeño de los modelos utilizando la matriz de confusión. Los verdaderos positivos son mostrados en la diagonal y representan las predicciones correctas, mientras que los valores fuera de la diagonal representan los errores cometidos por el modelo, acá están incluidos los falsos positivos y falsos negativos para cada clase. Una muestra de la información proporcionada por la matriz de confusión para KNeighbors Classifier se observa en la Tabla 28.

	0	1	16384	16385	32768	49152	65536	65537	81920	81921	86017	114688	147456	196608	212992	212993	213184	245760
0	76	0	0	0	0	1	2	0	1	0	0	0	0	11	0	0	0	0
1	0	16	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0
16384	0	0	210	0	0	1	0	0	1	0	0	0	0	0	12	0	0	0
16385	0	0	0	26	0	0	0	0	0	0	0	0	0	1	0	0	0	0
32768	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
49152	3	0	0	0	0	10	0	0	1	0	0	0	0	0	1	0	0	1
65536	3	0	0	0	0	0	36	0	1	0	0	0	0	5	0	0	0	1
65537	0	4	0	0	0	0	0	3	0	0	0	1	0	0	0	0	0	0
81920	2	0	0	0	0	0	1	0	49	0	0	0	1	1	1	0	0	0
81921	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
86017	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
114688	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	2
147456	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
196608	4	0	0	0	0	0	4	0	0	0	0	0	0	77	0	1	0	1
212992	1	0	4	1	0	0	0	0	2	0	0	0	0	0	129	1	0	1
212993	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0
213184	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
245760	2	0	0	0	0	2	0	0	0	0	0	3	0	2	0	0	0	2

Tabla 28: Matriz de confusión para KNeighbors Classifier

Las clases con pocas instancias muestran un rendimiento pobre debido a la sub representación de las clases minoritarias y el resultado puede ser engañoso debido a varios problemas como, el sesgo hacia las clases mayoritarias, la falta de generalización para las clases minoritarias y la pérdida de información relevante entre otros. Esto sugiere la necesidad de un análisis más detallado para abordar este desequilibrio. Inicialmente consideramos clases con más de cinco instancias, cuyo rendimiento se muestra en la Figura 65.

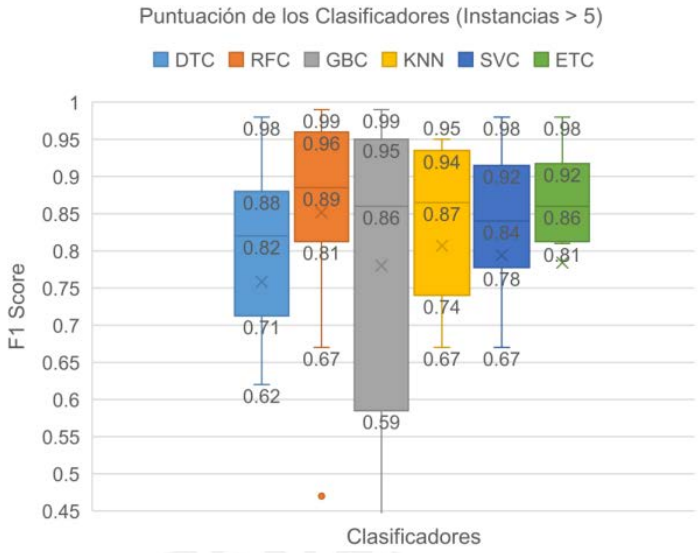


Figura 65. Puntuación de los clasificadores para grupos con más de 5 instancias

A partir de los resultados se evaluaron los clasificadores considerando su rendimiento, consistencia y costos computacionales en entornos con datos desbalanceados y multiclase. Los resultados evidencian que Random Forest Classifier (RFC) sobresale por su alta mediana de 0.89 y un intervalo intercuartílico (IQR) de 0.15, y lo presenta como una opción robusta y confiable, especialmente en situaciones de alta variabilidad en los datos. Sin embargo, comparado con Extra Trees Classifier (ETC) y Decision Tree Classifier (DTC), RFC requiere una mayor capacidad de memoria y procesamiento debido a su estructura basada en un número significativamente mayor de árboles y evaluaciones paralelas. Esto podría representar una limitante importante en plataformas con restricciones de hardware, donde ETC y DTC podrían ser opciones más eficientes en cuanto a recursos.

Por otro lado, Extra Trees Classifier (ETC) se posiciona como la opción más adecuada para entornos con recursos limitados, gracias a su equilibrio entre rendimiento y eficiencia computacional. Este clasificador registró una mediana de 0.86 y un IQR de 0.11, lo que refleja una buena estabilidad en sus predicciones. Su eficiencia computacional proviene de la reducida complejidad de su algoritmo, gracias a una menor cantidad de operaciones internas en la construcción de árboles, a diferencia de RFC y DTC que involucra más evaluaciones. Decision Tree Classifier (DTC), con una mediana de 0.82 y un IQR de 0.17, destaca por su simplicidad, rapidez e interpretabilidad, y lo hace particularmente útil en sistemas con restricciones severas de hardware. En comparación con Extra Trees Classifier (ETC) y Random Forest Classifier (RFC), DTC tiene una estructura más simple que requiere menos memoria y procesamiento, lo que resulta ventajoso en entornos desbalanceados y multiclase. Sin embargo, esta simplicidad también limita su capacidad para capturar patrones complejos en los datos, y puede llevar a un rendimiento inferior frente a ETC en cuanto a estabilidad y a RFC en cuanto a robustez. No obstante, su mayor propensión al sobreajuste debe ser gestionada cuidadosamente.

En cuanto a K-Nearest Neighbors (KNN), su mediana de 0.87 y un IQR de 0.2 lo posicionan como una alternativa viable en escenarios específicos donde se puedan optimizar sus altos requerimientos computacionales. Asimismo, Gradient Boosting Classifier (GBC) presenta un rendimiento potencial elevado con una mediana de 0.86, pero su IQR de 0.36 revela una alta variabilidad, lo que reduce su confiabilidad en comparación con otros clasificadores.

En síntesis, RFC se presenta como la mejor opción para este enfoque en escenarios donde los recursos computacionales no sean un impedimento, sin embargo, ETC se presenta como la opción más equilibrada para entornos con limitaciones de recursos, seguido de DTC por su simplicidad y eficiencia. Por su parte, KNN y GBC podrían considerarse en contextos específicos con estrategias de optimización adecuadas.

A continuación, se evaluó el desempeño de los clasificadores en clases con pocas instancias. Se utilizó la métrica F1-Score, adecuada para escenarios donde el balance entre precisión y exhaustividad es importante. Se consideraron los siguientes criterios:

Promedio: Un promedio más alto indica que el modelo tiende a tener métricas más altas en general.

Mediana: Es útil para identificar tendencias generales en presencia de valores extremos.

Máximo: Indica la calificación máxima alcanzada.

Proporción de valores válidos (Valid): Indica que porcentaje de los datos son No Cero.

La Tabla 29, muestra los resultados para Clases con menos de 6 instancias

		F1-Score					
Clase	Support	DTC	RFC	GBC	KNN	SVC	ETC
213184	5	0.6	0.89	0.67	0.73	0.89	0
32768	4	0.44	0	0.67	0	0.29	0.33
114688	4	0	0	0	0	0	0
147456	3	0.29	0.67	0.57	0.67	0.86	0.5
196800	3	0	0.33	0.57	0	0.57	0
196992	3	0.75	0.8	0.67	0.86	0.86	0
212993	3	0	0	0	0.33	0	0
86017	2	0	0.5	0.67	1	1	0
213376	1	0	0	0	0	0	0

Tabla 29. F1-Score para Clases con pocas instancias

El análisis comparativo de los modelos evaluados revela que SVC es el modelo más destacado, debido a su combinación de métricas superiores en términos de calidad y consistencia de los datos válidos. Este modelo presentó el promedio más alto (0.74), un máximo de rendimiento igual a 1.0 y un índice de proporción de valores válidos (Valid) de 0.67. Estos resultados indican que SVC mantiene un desempeño y consistencia superior. Por otro lado, KNN obtuvo resultados cercanos, con un promedio de 0.72 y un máximo de 1.0, aunque con un Valid ligeramente menor (0.56), lo que sugiere menor confiabilidad en sus datos. En contraste, GBC muestra un Valid elevado (0.67) pero presentó un máximo relativamente bajo (0.67), limitando su desempeño frente a los mejores modelos. Finalmente, los otros modelos, como DTC, RFC, y ETC, presentaron métricas inferiores tanto en promedio como en consistencia, que los ubica por debajo de SVC en términos de desempeño general. Con base en estos resultados, se concluye que SVC es el modelo más adecuado para grupos con pocas instancias en este entorno.

Finalmente, para validar el enfoque planteado, los algoritmos seleccionados se probaron de la siguiente manera: Se pre procesaron 240.000 muestras no vistas previamente, formando ventanas de tiempo de 90 segundos para el entrenamiento, y las 120.000 muestras posteriores se usaron para verificar las predicciones. El modelo se entrenó con 200 horas de datos muestreados y luego se usó para predecir 100 horas consecutivas. A partir del conjunto de datos de entrenamiento, se obtuvo una matriz de centroide de potencia para la codificación de etiquetas. Una vez que se entrenó el modelo, se realizaron predicciones sobre los datos no vistos. Los resultados revelaron inconsistencias en aproximadamente el 2% de los datos, donde las predicciones no correspondían a ninguna de las clases reales, probablemente debido a cambios en las condiciones de operación de la máquina que no se capturaron durante el entrenamiento. Los resultados finales, después de eliminar estas inconsistencias, se presentan en la Figura 66. RFC demostró ser el modelo con mejor rendimiento, con una puntuación F1-Score de 0,85 y un Recall de 0,86, lo que respalda los hallazgos planteados anteriormente.

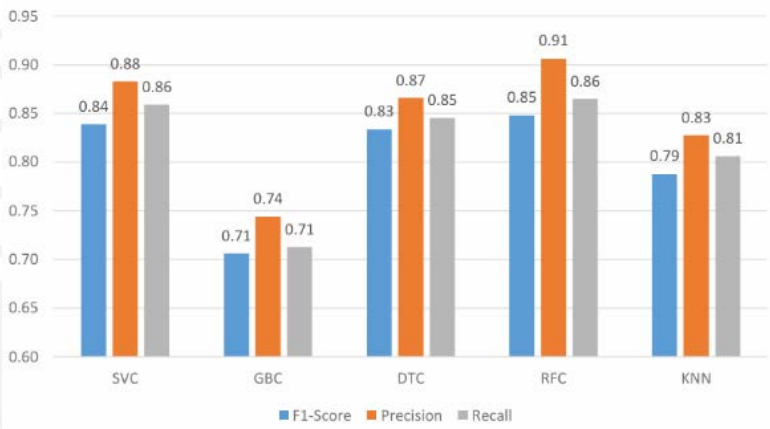


Figura 66. Puntuación obtenida con datos no observados previamente

5.8 Prueba del Modelo Entrenado Utilizando la Tarjeta de Evaluación Raspberry Pi

Para realizar las pruebas en la tarjeta de evaluación se requiere un sistema que comprende las partes mostrada en la Figura 67.

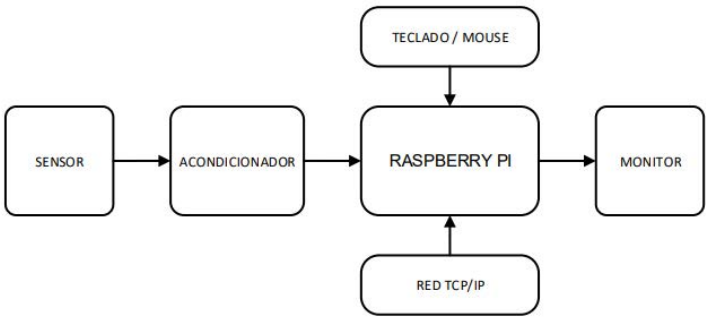


Figura 67. Diagrama de bloques del sistema con Raspberry

Recursos de Hardware: Para el desarrollo y pruebas, se utilizaron los siguientes recursos de hardware:

Computadora Personal con las siguientes características:

- Procesador: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz - 2.90 GHz.
- Memoria RAM: 12 GB.
- Sistema Operativo: 64 bits.
- Almacenamiento: SSD de 1 TB.

Tarjeta de Evaluación Raspberry Pi 4:

- Procesador: Broadcom BCM2711, 64 bits @ 1.8GHz.
- Memoria: 4 GB SDRAM.
- Sistema Operativo: Raspberry Pi OS.
- Almacenamiento: Micro-SD de 32 GB.

Los datos de entrada son señales de corriente y tensión muestreadas a una frecuencia de 1 Hz, las cuales ingresan a un circuito de acondicionamiento antes de su procesamiento en la Raspberry Pi.

Entrenamiento y Preparación del Modelo: El entrenamiento del modelo se realizó en Google Colaboratory (Colab), un entorno en la nube que permite la ejecución de código Python con acceso a recursos avanzados como CPU y GPU. Se configuró el entorno con los siguientes parámetros:

- Recursos: CPU con 12 GB de RAM y disco asignado de 100 GB.
- Almacenamiento de datos se realizó en Google Drive.
- Configuración del Entorno: Python 3.10.12 con las siguientes bibliotecas:
  NumPy (versión 1.26.4).
  Pandas (versión 2.2.2).
  Scikit-learn (versión 1.6.0).

El uso de Colab permitió aprovechar recursos avanzados, aunque estuvo limitado por la duración de las sesiones gratuitas. El modelo seleccionado fue Random Forest Classifier (RFC), que demostró un equilibrio óptimo entre precisión y rendimiento. El modelo fue exportado como un archivo pickle (.pkl) para su posterior implementación.

El procedimiento en la Raspberry Pi incluyó los siguientes pasos:

Configuración de la Raspberry Pi: Se instalaron todas las dependencias necesarias para ejecutar el modelo exportado. El modelo pkl fue copiado a la tarjeta y preparado para su ejecución.

Procesamiento de Datos: Los datos de entrada provinieron de la base de datos REED, la cual contiene lecturas de potencia activa y tiempo. Se realizó el pre procesamiento de los datos para asegurar su compatibilidad con el modelo entrenado en Colab.

Inferencia del Modelo: La inferencia se llevó a cabo utilizando los datos pre procesados. Se midieron los tiempos de pre procesamiento e inferencia durante la ejecución del modelo en la Raspberry Pi, obteniendo un tiempo de 61 segundos, que demuestran la viabilidad del enfoque propuesto para plataformas de bajo costo.

CONCLUSIONES

El objetivo principal que fue diseñar y evaluar un enfoque para la medición de energía eléctrica no intrusiva fue alcanzado con éxito. Se desarrolló un sistema que integra técnicas de aprendizaje de máquinas con un método de codificación de datos, demostrando la viabilidad del enfoque propuesto en entornos con recursos limitados.

La revisión literaria permitió identificar las técnicas más relevantes de aprendizaje de máquinas aplicables a sistemas de medición no intrusiva, destacando especialmente los modelos basados en árboles de decisión. Estas técnicas fueron seleccionadas y probadas en el contexto del estudio para medir su efectividad.

El software desarrollado implementó las técnicas seleccionadas, utilizando la base de datos REED para entrenamiento y validación. El sistema, que incluyó procesos de extracción de características basados en firmas de potencia promedio, fue validado en un entorno controlado, demostrando su capacidad para realizar inferencias con buena precisión.

Se evaluaron diversos algoritmos de aprendizaje de máquinas, y el análisis comparativo identificó al algoritmo Random Forest como el más adecuado para el enfoque propuesto. Este modelo logró un balance óptimo entre precisión, tiempo de inferencia y facilidad de implementación, convirtiéndose en la opción más eficaz para el propósito de la investigación.

El modelo propuesto, Random Forest Classifier fue probado satisfactoriamente en una Raspberry Pi 4, confirmando su viabilidad para su implementación en plataformas de bajo costo y capacidades limitadas.

Se propuso una técnica de reducción de subestados mediante un enfoque de filtrado para eliminar aquellos con menor impacto en la determinación del consumo de energía. Esta estrategia mantuvo una buena aproximación a las lecturas reales de potencia, lo que permitió reducir la complejidad del modelo y del proceso de entrenamiento. Además, la implementación de codificación binaria para la formación de etiquetas facilitó la representación de los subestados del sistema eléctrico

Los modelos de aprendizaje automático, como Support Vector Classifier (SVC) y el Gradient Boosting Classifier (GBC), demostraron ser efectivos para clasificar clases minoritarias.

Los resultados obtenidos confirman que el enfoque propuesto es efectivo para la medición de energía eléctrica mediante el Monitoreo No Intrusivo de Cargas (NILM), a pesar de algunos sesgos en la precisión de los valores de potencia.

RECOMENDACIONES

Se propone incluir datos no eléctricos como temperatura ambiental, consumo de agua o presencia entre otros para mejorar el entrenamiento de los modelos de clasificación, esto permitirá la identificación de patrones de uso de aparatos eléctricos.

Se recomienda la incorporación de datos de potencia reactiva como una característica adicional para el proceso de clasificación, esto aportará información que permite distinguir entre cargas con perfiles similares en términos de potencia activa.

Se sugiere investigar técnicas de codificación más eficientes que permitan minimizar la cantidad de bits requeridos para representar los estados del sistema con el objetivo de reducir la complejidad computacional y el uso de memoria en los sistemas embebidos.

Se propone complementar el uso del algoritmo K-Means con herramientas estadísticas, para mejorar la caracterización de los estados de los aparatos eléctricos de consumo con el objetivo de estimar de manera más precisa la potencia eléctrica.

Evaluar otras técnicas para reducir la dimensionalidad, a fin de evitar una pérdida significativa de información relevante y mantener la representatividad de los datos, especialmente cuando se trata de datos multiclase y desbalanceado.

Se sugiere que los fabricantes de aparatos eléctricos incluyan fichas técnicas con curvas de ensayo representativos para facilitar el rentrenamiento de los modelos, principalmente cuando se añadan nuevos aparatos eléctricos a la red.

Finalmente, se sugiere que futuras investigaciones se dirijan hacia el uso de sistemas más robustos, como el aprendizaje profundo con arquitecturas compactas optimizadas para dispositivos de bajos recursos, para estimar de manera más preciso el consumo energético.

REFERENCIAS BIBLIOGRÁFICAS

[1] IPCC, 2018: Global Warming of 1.5°C.An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty [Masson-Delmotte, V., P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P.R. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R. Pidcock, S. Connors, J.B.R. Matthews, Y. Chen, X. Zhou, M.I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, and T. Waterfield (eds.)]. In Press.

[2] International Energy Agency. (2018). *Global energy and CO2 status report – 2017.* IEA.

[3] Banco Mundial. (n.d.). Combustibles fósiles (% del uso total de energía). *Datos del Banco Mundial.* Recuperado el 8 de enero de 2025, de https://datos.bancomundial.org/indicador/EG.USE.COMM.FO.ZS

[4] G. W. Hart, "Nonintrusive appliance load monitoring," in Proceedings of the IEEE, vol. 80, no. 12, pp. 1870-1891, Dec. 1992, doi: 10.1109/5.192069.

[5] Kwangduk Douglas Lee, Electric Load Information System based on Non-Intrusive Power Monitoring. Massachusetts Institute of Technology June 2003.

[6] Zoha, A.; Gluhak, A.; Imran, M.A.; Rajasegarar, S. Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey. Sensors 2012, 12, 16838-16866. https://doi.org/10.3390/s121216838

[7] Ruano, A.; Hernandez, A.; Ureña, J.; Ruano, M.; Garcia, J. NILM Techniques for Intelligent Home Energy Management and Ambient Assisted Living: A Review. Energies 2019, 12, 2203. https://doi.org/10.3390/en12112203.

[8] Zhuang, Mengmeng & Shahidehpour, M. & Li, Zuyi. (2018). An Overview of Non-Intrusive Load Monitoring: Approaches, Business Applications, and Challenges. 4291-4299. 10.1109/POWERCON.2018.8601534.

[9] Christoforos Nalmpantis and Dimitris Vrakas. 2019. Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparation. Artif. Intell. Rev. 52, 1 (June 2019), 217–243. https://doi.org/10.1007/s10462-018-9613-7

[10] Batra, N., Kelly, J., Parson, O., Dutta, H., Knottenbelt, W., Rogers, A., Singh, A., & Srivastava, M. (2014, junio). *NILMTK: An open source toolkit for non-intrusive load monitoring*. En *Proceedings of the 5th International Conference on Future Energy Systems (e-Energy '14)* (pp. XX–XX). ACM. https://doi.org/10.1145/2602044.2602051

[11] Makonin, Stephen. (2014). Real-Time Embedded Low-Frequency Load Disaggregation.

[12] Kolter, J & Johnson, Matthew. (2011). REDD: A Public Data Set for Energy Disaggregation Research. Artif. Intell.. 25.

[13] Le, Quoc & Ranzato, Marc'Aurelio & Monga, Rajat & Devin, Matthieu & Chen, Kai & Corrado, G.s & Dean, Jeff & Ng, Andrew. (2011). Building high-level features using large scale unsupervised learning. Proceedings of ICML. 1.

[14] Leslie K. Norford, Steven B. Leeb, Non-intrusive electrical load monitoring in comercial buildings based on steady-state and transient load-detection algorithms, Energy and Buildings, Volume 24, Issue 1, 1996, Pages 51-64, ISSN 0378-7788, https://doi.org/10.1016/0378-7788(95)00958-2.

[15] Farinaccio, Linda & Zmeureanu, R.. (1999). Using pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. Energy and Buildings. 30. 245-259. 10.1016/S0378-7788(99)00007-9.

[16] S. Drenker and A. Kader, "Nonintrusive monitoring of electric loads," in IEEE Computer Applications in Power, vol. 12, no. 4, pp. 47-51, Oct. 1999, doi: 10.1109/67.795138.

[17] Ruzzelli, A.G.; Nicolas, C.; Schoofs, A.; O'Hare, G.M.P. Real-Time Recognition and Profiling of Appliances through a Single Electricity Sensor. In Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, Boston, MA, USA, 21–25 June 2010; pp. 1–9.

[18] Y. -C. Su, K. -L. Lian and H. -H. Chang, "Feature Selection of Non-intrusive Load Monitoring System Using STFT and Wavelet Transform," *2011 IEEE 8th International Conference on e-Business Engineering*, Beijing, China, 2011, pp. 293-298, doi: 10.1109/ICEBE.2011.49.

[19] Lam, H.Y. & Fung, G.S.K. & Lee, W.K.. (2007). A Novel Method to Construct Taxonomy Electrical Appliances Based on Load Signaturesof. Consumer Electronics, IEEE Transactions on. 53. 653 - 660. 10.1109/TCE.2007.381742.

[20] Chang, Hsueh-Hsien & Yang, Hong-Tzer & Lin, Ching-Lung. (2008). Load Identification in Neural Networks for a Non-intrusive Monitoring of Industrial Electrical Loads. Lecture Notes in Computer Science. 5236. 664-674.

[21] A. Cole and A. Albicki, "Nonintrusive identification of electrical loads in a three-phase environment based on harmonic content," Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference [Cat. No. 00CH37066], Baltimore, MD, USA, 2000, pp. 24-29 vol.1, doi: 10.1109/IMTC.2000.846806.

[22] K. D. Lee, S. B. Leeb, L. K. Norford, P. R. Armstrong, J. Holloway and S. R. Shaw, "Estimation of variable-speed-drive power consumption from harmonic content," in IEEE Transactions on Energy Conversion, vol. 20, no. 3, pp. 566-574, Sept. 2005, doi: 10.1109/TEC.2005.852963.

[23] W. Wichakool, A. -T. Avestruz, R. W. Cox and S. B. Leeb, "Modeling and Estimating Current Harmonics of Variable Electronic Loads," in IEEE Transactions on Power Electronics, vol. 24, no. 12, pp. 2803-2811, Dec. 2009, doi: 10.1109/TPEL.2009.2029231.

[24] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. 2010. ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home. In Proceedings of the 12th ACM international conference on Ubiquitous computing (UbiComp '10). Association for Computing Machinery, New York, NY, USA, 139–148. https://doi.org/10.1145/1864349.1864375

[25] Guvensan M.A., Taysi Z.C., Melodia T. Energy monitoring in residential spaces with audio sensor nodes: TinyEARS. Ad Hoc Netw. 2013;11:1539–1555. doi: 10.1016/j.adhoc.2012.10.002

[26] Brown, R., Ghavami, N., Siddiqui, H., Adjrad, M., Ghavami, M. and Dudley-Mcevoy, S. (2017). Occupancy based household energy disaggregation using ultra wideband radar and electrical signature profiles. Energy and Buildings. 141, pp. 134-141. https://doi.org/10.1016/j.enbuild.2017.02.004

[27] Egarter, D., Sobe, A., Elmenreich, W. (2013). Evolving Non-Intrusive Load Monitoring. In: Esparcia-Alcázar, A.I. (eds) Applications of Evolutionary Computation. EvoApplications 2013. Lecture Notes in Computer Science, vol 7835. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37192-9_19

[28] Suzuki, Kosuke & Inagaki, Shinkichi & Suzuki, Tatsuya & Nakamura, Hisahide & Ito, Koichi. (2008). Nonintrusive Appliance Load Monitoring Based on Integer Programming. Ieej Transactions on Power and Energy. 128. 2742 - 2747. 10.1109/SICE.2008.4655131.

[29] Bhotto, M.Z.A., Makonin, S., Bajic, I.: Load disaggregation based on aided linear integer programming. IEEE Trans. Circuits Syst. II Express Briefs 1 (2016). doi:10. 1109/TCSII.2016.2603479. ISSN: 1549-7747

[30] Kong, W., Dong, Z. Y., Hill, D. J., Luo, F., & Xu, Y. (2016). Improving nonintrusive load monitoring efficiency via a hybrid programing method. *IEEE Transactions on Industrial Informatics*, *12*(6), 2148-2157. https://doi.org/10.1109/TII.2016.2590359

[31] Wong, Y. F., Sekercioglu, Y. A., Drummond, T. W., & Wong, V. S. (2013). Recent approaches to non-intrusive load monitoring techniques in residential settings. In S. Sundaram (Ed.), *Proceedings of the 2013 IEEE Computational Intelligence Applications in Smart Grid (CIASG)* (pp. 73 - 79). IEEE, Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/CIASG.2013.6611501

[32] Kim H, Marwah M, Arlitt MF, Lyon G, Han J (2011) Unsupervised disaggregation of low frequency power measurements. In: Proceedings of the 2011 SIAM international conference on data mining, vol 11, pp 747–758 https://doi.org/10.1137/1.9781611972818.64

[33] Kolter, J.Z.; Jaakkola, T. (2012). Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. - Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, in Proceedings of Machine Learning Research 22:1472-1482

[34] Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers. 2012. Non-intrusive load monitoring using prior models of general appliance types. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12). AAAI Press, 356–362.

[35] M. J. Johnson and A. S. Willsky, “Bayesian nonparametric hidden semi-markov models,” The Journal of Machine Learning Research, vol. 14, no. 1, pp. 673–701, 2013. 39, 40, 42, 81

[36] Zia, Tehseen & Bruckner, Dietmar & Zaidi, Adeel. (2011). A Hidden Markov Model Based Procedure for Identifying Household Electric Loads. IECON Proceedings (Industrial Electronics Conference). 10.1109/IECON.2011.6119826.

[37] Zeifman, Michael & Roth, Kurt. (2011). Viterbi algorithm with sparse transitions (VAST) for nonintrusive load monitoring. Proceedings of IEEE Symposium Series in Computational Intelligence. 10.1109/CIASG.2011.5953328.

[38] Figueiredo, Marisa & De Almeida, Ana & Ribeiro, Bernardete. (2011). An Experimental Study on Electrical Signature Identification of Non-Intrusive Load Monitoring (NILM) Systems. 31-40. 10.1007/978-3-642-20267-4_4.

[39] Suman Giri, Mario Bergés, Anthony Rowe, Towards automated appliance recognition using an EMF sensor in NILM platforms, Advanced Engineering Informatics, Volume 27, Issue 4, 2013, Pages 477-485, ISSN 1474-0346, https://doi.org/10.1016/j.aei.2013.03.004.

[40] Srinivasan, D. & Ng, W.S. & Liew, A.C.. (2006). Neural-Network-Based Signature Recognition for Harmonic Source Identification. Power Delivery, IEEE Transactions on. 21. 398 - 405. 10.1109/TPWRD.2005.852370.

[41] Lin, Gu-Yuan & Lee, Shih-chiang & Hsu, Jane & Jih, Wan-Rong. (2010). Applying power meters for appliance recognition on the electric panel. Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications, ICIEA 2010. 2254 - 2259. 10.1109/ICIEA.2010.5515385.

[42] Lai, Ying-Hsun & Lai, Chin-Feng & Huang, Yueh-Min. (2013). Multi-appliance recognition system with hybrid SVM/GMM classifier in ubiquitous smart home. Information Sciences. 230. 39–55. 10.1016/j.ins.2012.10.002.

[43] Mendes Lemes, Dimas & Cabral, Thales & Fraidenraich, G. & Meloni, Luis & De Lima, Eduardo & Neto, Fernando. (2021). Load Disaggregation Based on Time Window for HEMS Application. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3078340.

[44] Moreno Jaramillo, Andres Felipe & Laverty, D. & Martinez-del-Rincon, Jesus & Hastings, John & Morrow, D.J.. (2020). Supervised Non-Intrusive Load Monitoring Algorithm for Electric Vehicle Identification. 1-6. 10.1109/I2MTC43012.2020.9128529.

[45] Bosco, Thais & Serrão Gonçalves, Flávio & Souza, Wesley. (2021). A Comparative Study of Machine Learning Classifiers for Electric Load Disaggregation based on an extended NILM dataset. 270-277. 10.1109/INDUSCON51756.2021.9529824.

[46] Meehan, P., McArdle, C., & Daniels, S. (2014). An Efficient, Scalable Time-Frequency Method for Tracking Energy Usage of Domestic Appliances Using a Two-Step Classification Algorithm. *Energies*, *7*(11), 7041-7066. https://doi.org/10.3390/en7117041

[47] Barker, Sean & Musthag, Mohamed & Irwin, David & Shenoy, Prashant. (2015). Non-intrusive load identification for smart outlets. 2014 IEEE International Conference on Smart Grid Communications, SmartGridComm 2014. 548-553. 10.1109/SmartGridComm.2014.7007704.

[48] He, H., Liu, Z., Jiao, R., & Yan, G. (2019). A Novel Nonintrusive Load Monitoring Approach based on Linear-Chain Conditional Random Fields. *Energies*, *12*(9), 1797. https://doi.org/10.3390/en12091797

[49] C. Zhou, S. Liu and P. Liu, "Neural Network Pattern Recognition Based Non-intrusive Load Monitoring for a Residential Energy Management System," 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), Beijing, China, 2016, pp. 483-487, doi: 10.1109/ICISCE.2016.110.

[50] T. Sirojan, B. T. Phung and E. Ambikairajah, "Deep Neural Network Based Energy Disaggregation," 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 2018, pp. 73-77, doi: 10.1109/SEGE.2018.8499441.

[51] A. Kundu, G. P. Juvekar and K. Davis, "Deep Neural Network Based Non-Intrusive Load Status Recognition," 2018 Clemson University Power Systems Conference (PSC), Charleston, SC, USA, 2018, pp. 1-6, doi: 10.1109/PSC.2018.8664063.

[52] Weiwei, Miao & Zeng, Zeng & Changzhi, Teng & Sibo, Bi & Rui, Zhang & Shihao, Li. (2022). Non-invasive Load Identification Method Based on the Characteristics of Residential Electrical Appliances. 827-832. 10.1109/CEEPE55110.2022.9783314.

[53] Prasertlux, Ruttagorn & Sudwilai, Phaisarn & Budsabathon, Chatree. (2022). Design and Development of Smart Meter Load Profile for Residences. 743-747. 10.1109/ICBIR54589.2022.9786393.

[54] J. Liang, S. K. K. Ng, G. Kendall and J. W. M. Cheng, "Load Signature Study—Part I: Basic Concept, Structure, and Methodology," in IEEE Transactions on Power Delivery, vol. 25, no. 2, pp. 551-560, April 2010, doi: 10.1109/TPWRD.2009.2033799.

[55] Chowdhury, D., Hasan, M., and Rahman Khan, M.Z. (2020, January 28–29). Statistical Features Extraction from Current Envelopes for Non-Intrusive Appliance Load Monitoring. Proceedings of the 2020 SoutheastCon, Raleigh, NC, USA. https://doi.org/10.1109/SoutheastCon44009.2020.9249667

[56] Mitchell, T. M. (1997). Machine Learning (Ed. ilustrada). McGraw-Hill.

[57] Dabbura, I. (2018, September 17). *K-means clustering: Algorithm, applications, evaluation methods, and drawbacks*. Towards Data Science. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

[58] S. Abokadr, A. Azman, H. Hamdan and N. Amelina, "Handling Imbalanced Data for Improved Classification Performance: Methods and Challenges," 2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA), Taiz, Yemen, 2023, pp. 1-8, doi: 10.1109/eSmarTA59349.2023.10293442.

[59] Puente Águeda, Cristina & Palacios, Rafael & González-Arechavala, Yolanda & Sánchez, Eugenio. (2020). Non-Intrusive Load Monitoring (NILM) for Energy Disaggregation Using Soft Computing Techniques. Energies. 13. 3117. 10.3390/en13123117.

[60] A. Deshmukh and D. Lohan, Electric Load Identification using Machine Learning, CS446 Project Report, University of Illinois, May 2015.

[61] A. U. Rehman, T. T. Lie, B. Vallès and S. R. Tito, "Comparative Evaluation of Machine Learning Models and Input Feature Space for Non-intrusive Load Monitoring," in Journal of Modern Power Systems and Clean Energy, vol. 9, no. 5, pp. 1161-1171, September 2021, doi: 10.35833/MPCE.2020.000741.

ANEXO A

Este anexo presenta el comportamiento de los aparatos eléctricos basado en los datos REED. Se incluyen las siguientes figuras: (1) la curva del codo, que muestra el número óptimo de clústeres, (2) los centroides, que representan patrones de carga, y (3) las formas de onda de las cargas medidas en house_3.

Tomacorriente 3. De la Figura A1-1 curva del codo sugiere 4 clusters. La Figura A1-2 identifica los centroides de los clusters con las potencias representativas 79 W, 1259 W, 152 W, 101 W. La Figura A1-3 muestra la ubicación de las potencias promedio en la representación temporal.

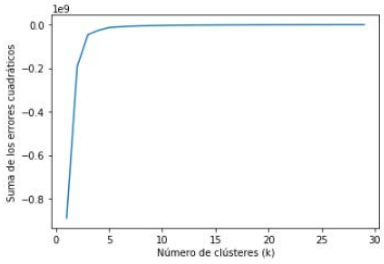


Figura A1-1. Gráfica de Codo para Tomacorriente 3

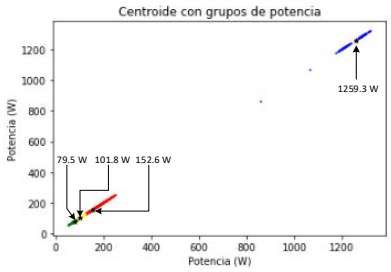


Figura A1-2. Gráfica de Centroides para Tomacorriente 3

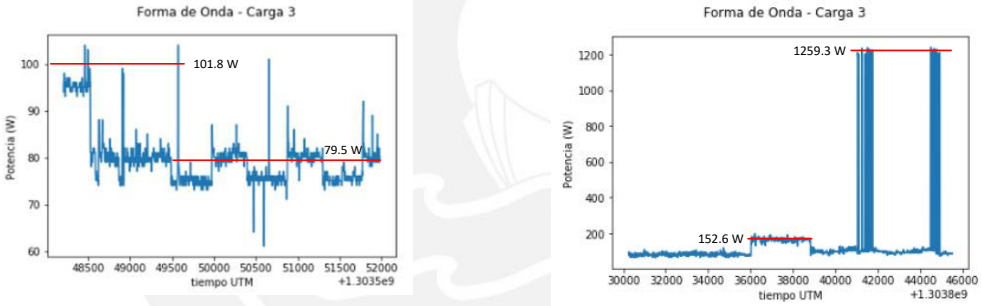


Figura A1-3. Gráfica de potencia en el tiempo para Tomacorriente 3

Tomacorriente 4. De la Figura A2-1, curva del codo sugiere 4 clusters. La Figura A2-2 identifica los centroides de los clusters con las potencias representativas 0.41W, 118.9W, 405.7W, 1312.9W. La Figura A2-3 muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

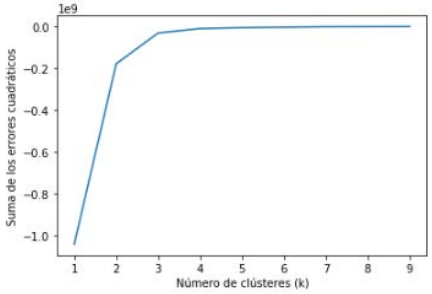


Figura A2-1. Gráfica de Codo para Tomacorriente 4

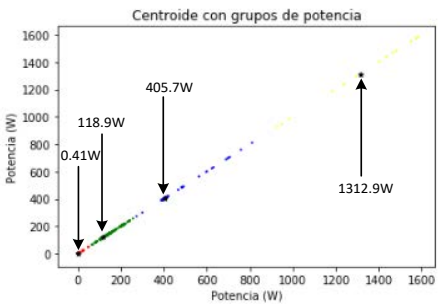


Figura A2-2. Gráfica de Centroides para Tomacorriente 4

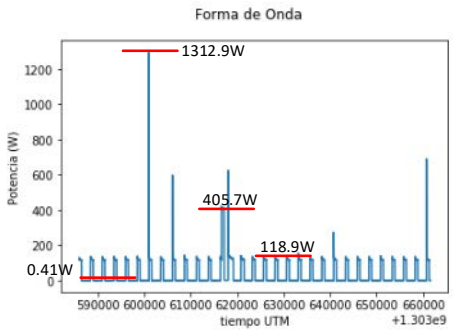


Figura A2-3. Gráfica de potencia en el tiempo para Tomacorriente 4

Aparato electrónico 6. De la Figura A3-1. curva del codo sugiere 4 clusters. La Figura A3-2. identifica los centroides de los clusters con las potencias representativas 0.9 W, 729 W, 208.9 W, 131.5 W. La Figura A3-3. muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

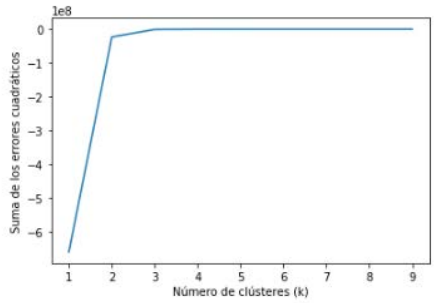


Figura A3-1. Gráfica de Codo para Aparato electrónico 6

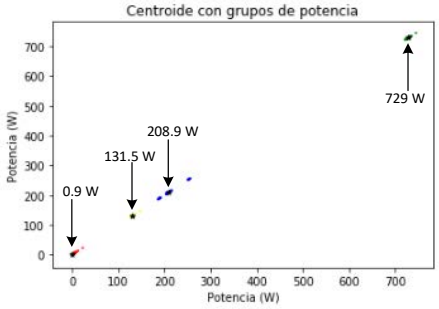


Figura A3-2. Gráfica de Centroides para Aparato electrónico 6

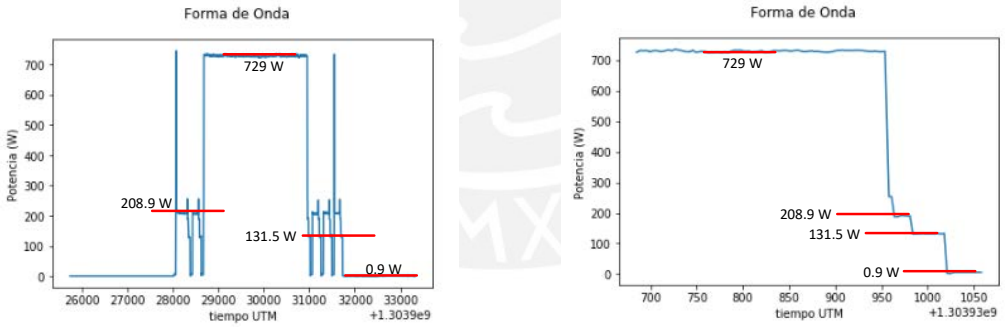


Figura A3-3. Gráfica de potencia en el tiempo para Aparato electrónico 6

Refrigerador 7. De la Figura A4-1, curva del codo sugiere 4 clusters. La Figura A4-2 identifica los centroides de los clusters con las potencias representativas 5 W, 718.4 W, 581.8 W, 111.1 W. La Figura A4-3. muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

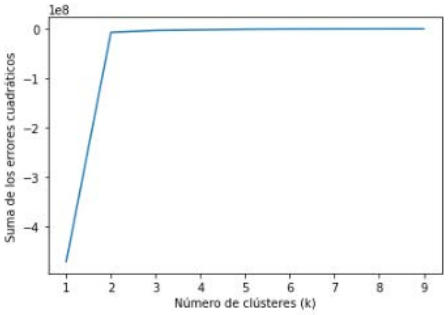


Figura A4-1. Gráfica de Codo para Refrigerador 7

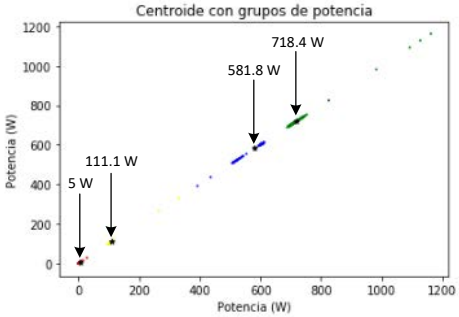


Figura A4-2. Gráfica de Centroides para Refrigerador 7

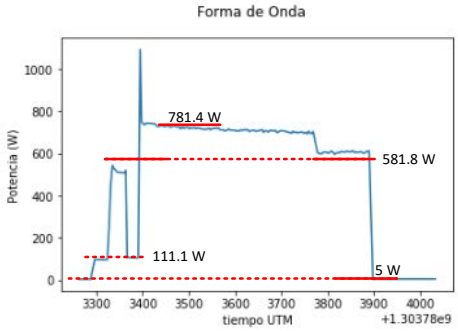


Figura A4-3. Gráfica de potencia en el tiempo para Refrigerador 7

Horno 10. De la Figura A5-1, curva del codo sugiere 2 clusters, en este caso consideramos 4 clusters que se adecúa a la codificación propuesta. La Figura A5-2 identifica los centroides de los clusters con las potencias representativas 0 W, 2296 W, 2497 W, 2238 W. La Figura A5-3. muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

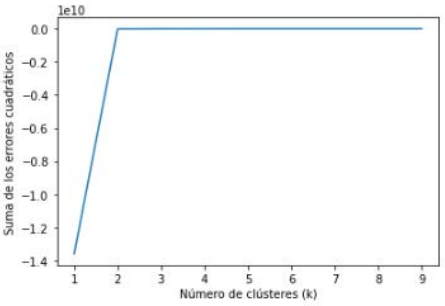


Figura A5-1. Gráfica de Codo para Horno 10

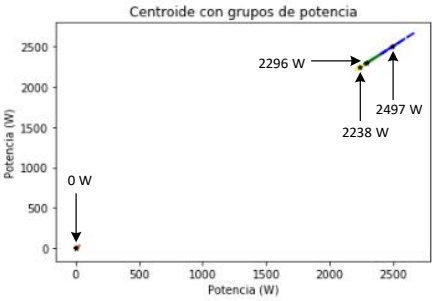


Figura A5-2. Gráfica de Centroides para Horno 10

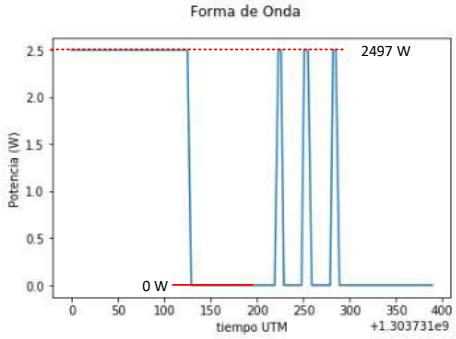


Figura A5-3. Gráfica de potencia en el tiempo para Horno 10

Iluminación 11. De la Figura A6-1, curva del codo sugiere 3 clusters, en este caso consideramos 4 clusters que se adecúa a la codificación propuesta. La Figura A6-2 identifica los centroides de los clusters con las potencias representativas 0 W, 2513 W, 267.2 W, 2740 W. La Figura A6-3 muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

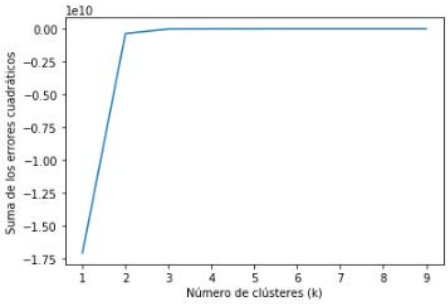


Figura A6-1. Gráfica de Codo para Iluminación 11

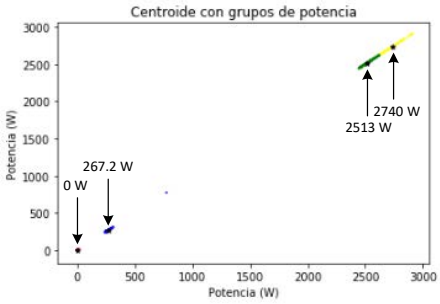


Figura A6-2. Gráfica de Centroides para Iluminación 11

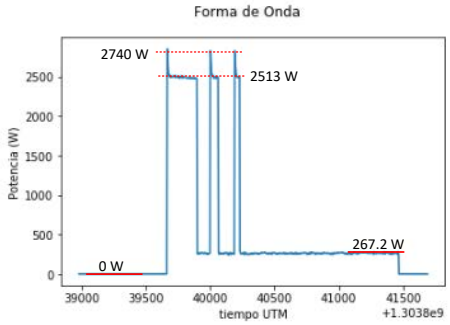


Figura A6-3. Gráfica de potencia en el tiempo para Iluminación 11

Lavadora 13. De la Figura A7-1, curva del codo sugiere 2 clusters, en este caso consideramos 4 clusters que se adecúa a la codificación propuesta. La Figura A7-2 identifica los centroides de los clusters con las potencias representativas 2 W, 1718 W, 124.4 W, 391.9 W. La Figura A7-3 muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

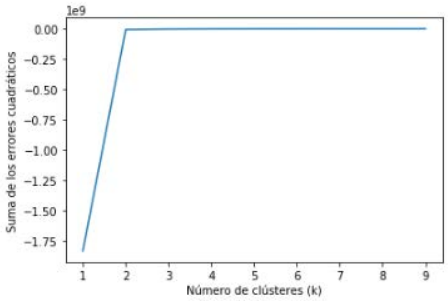


Figura A7-1. Gráfica de Codo para Lavadora 13

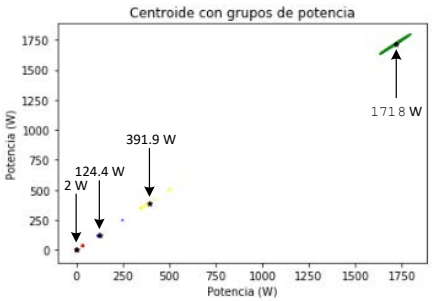


Figura A7-2. Gráfica de Centroides para Lavadora 13

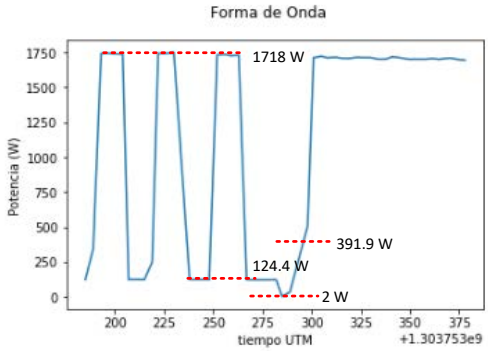


Figura A7-2. Gráfica de potencia en el tiempo para Lavadora 13

Iluminación 17. De la Figura A8-1, curva del codo sugiere 2 clusters, en este caso consideramos 4 clusters que se adecúa a la codificación propuesta. La Figura A8-2 identifica los centroides de los clusters con las potencias representativas 0 W, 1290 W, 1595 W, 948 W. La Figura A8-3 muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

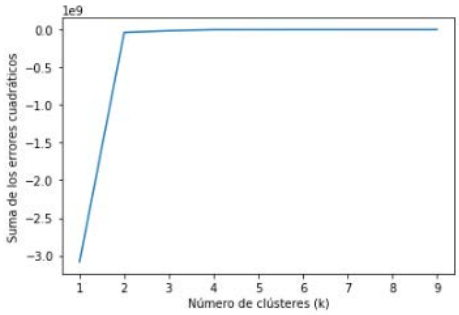


Figura A8-1. Gráfica de Codo para Iluminación 17

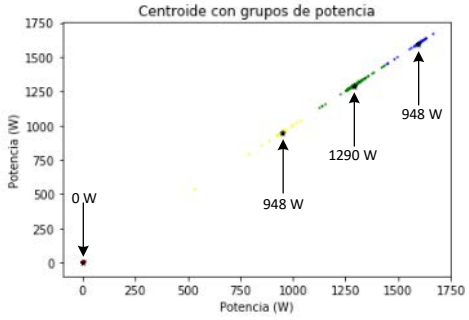


Figura A8-2. Gráfica de Centroides para Iluminación 17

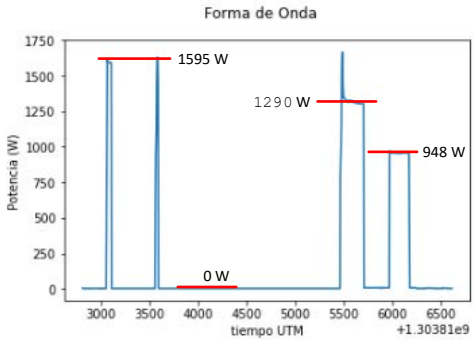


Figura A8-3. Gráfica de potencia en el tiempo para Iluminación 17

Iluminación 19. De la Figura A9-1, curva del codo sugiere 5 clusters. La Figura A9-2 muestra los sub estados considerado 4 y 5 clusters. Para una codificación de 2 bits consideramos 4 clusters. La Figura 35-b identifica los centroides de los clusters con las potencias representativas 2 W, 130 W, 366 W, 1038 W. La figura A9-3 muestra la ubicación de la potencia promedio en la representación temporal de la potencia.

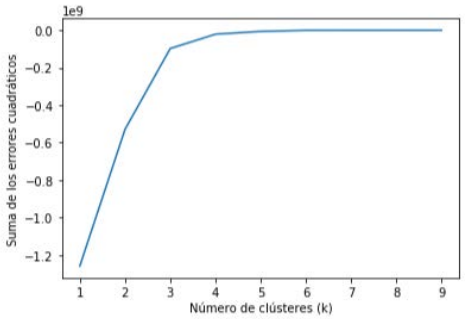


Figura A9-1. Gráfica de Codo para Iluminación 19

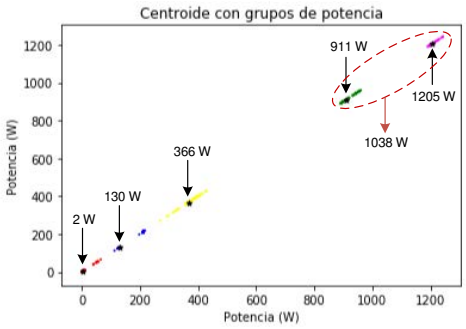


Figura A9-2. Gráfica de Centroides para Iluminación 19

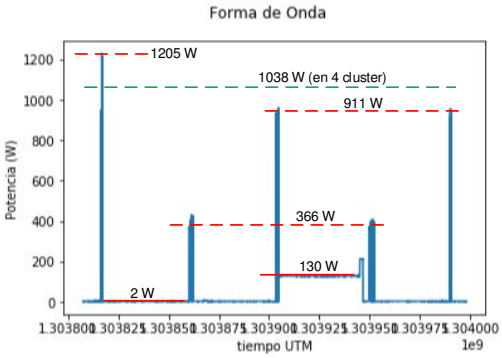


Figura A9-3. Gráfica de potencia en el tiempo para Iluminación 19

ANEXO B

Curvas de validación de los modelos seleccionados para evaluar su rendimiento en función de los parámetros configurados.

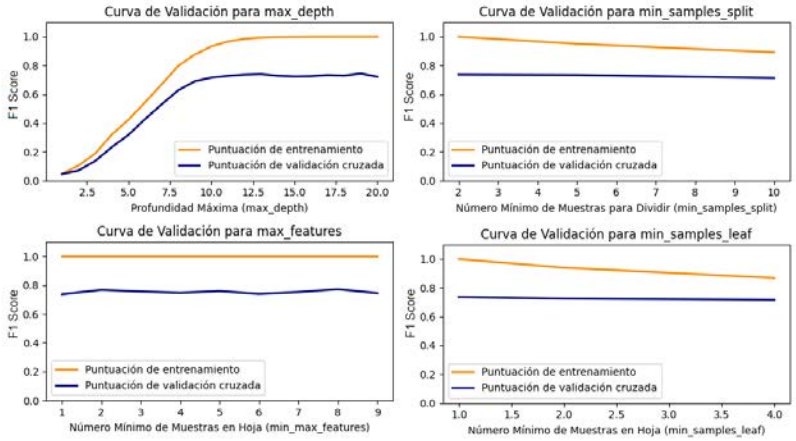


Figura B1. Curvas de Validación Decision Tree Classifier

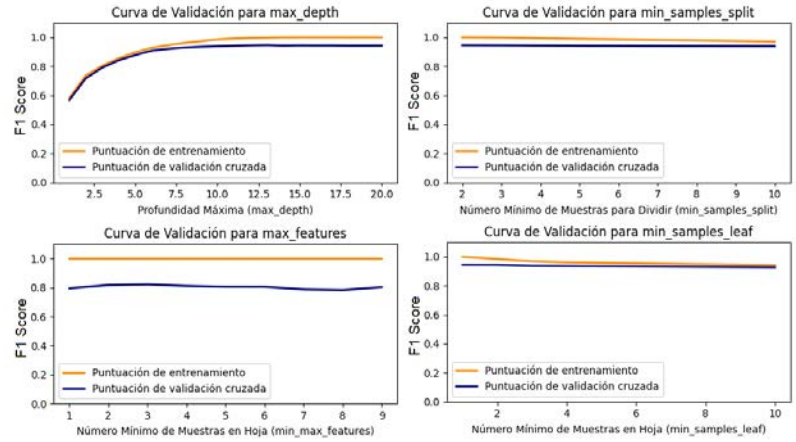


Figura B2. Curvas de Validación Gradient Boosting Classifier

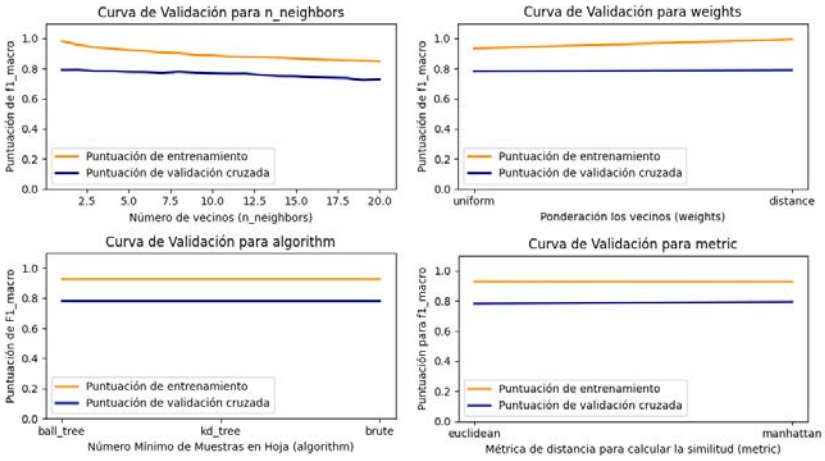


Figura B3. Curvas de Validación K-Neighbors Classifier

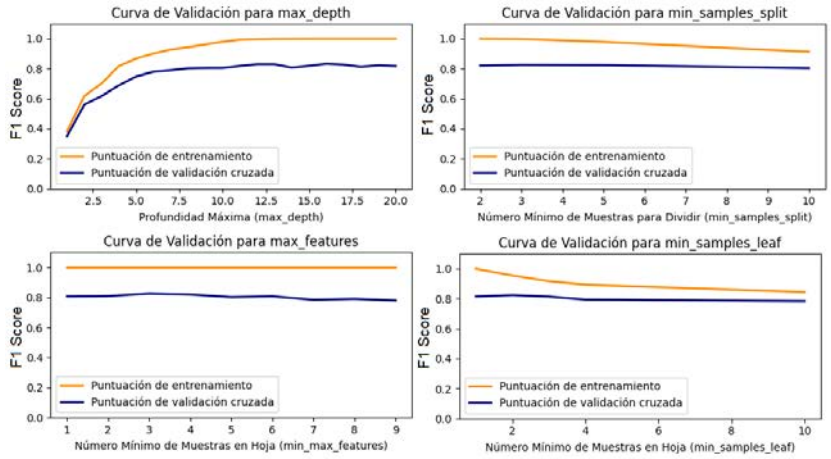


Figura B4. Curvas de Validación Random Forest Classifier

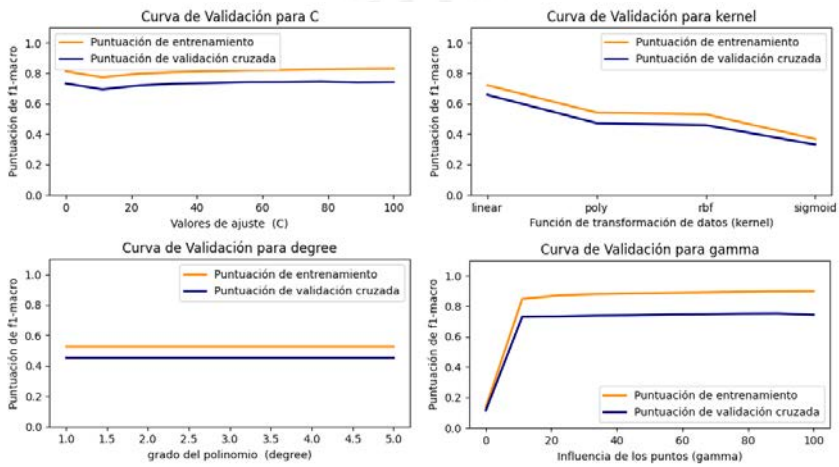


Figura B5. Curvas de Validación Support Vector Classifier

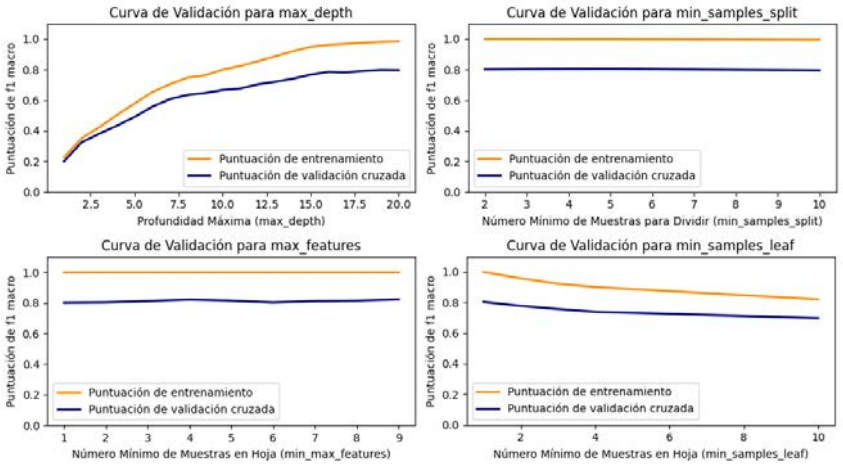


Figura B6. Curvas de Validación Extra Trees Classifier

ANEXO C

Matriz de Confusión de los modelos entrenados.

PREDICCION

REAL	0	1	16384	16385	32768	49152	65536	65537	81920	81921	86017	114688	147456	196608	196800	196992	212992	212993	213184	213376	245760
0	63	0	0	2	0	3	0	0	2	0	0	0	0	9	0	0	0	0	0	0	0
1	0	20	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0
16384	0	0	211	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
16385	0	0	0	20	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
32768	0	0	0	0	2	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
49152	1	0	1	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
65536	0	2	0	0	0	0	46	0	0	1	0	1	1	1	0	0	0	0	0	0	0
65537	0	2	0	0	1	0	0	1	0	0	2	0	0	0	0	0	0	0	2	0	0
81920	1	0	0	0	0	0	2	0	47	0	0	1	1	1	0	0	2	0	0	0	1
81921	0	1	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
86017	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
114688	0	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0
147456	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0
196608	7	0	0	7	0	0	2	0	2	0	0	0	0	71	0	0	0	0	0	0	0
196800	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
196992	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
212992	0	0	4	0	0	1	0	0	1	0	0	0	0	0	0	0	150	0	0	0	0
212993	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
213184	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0	0
213376	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
245760	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5

Figura C1. Matriz de Confusión del Modelo Decision Tree Classifier (DTC)

PREDICCIÓN

REAL	0	1	16384	16385	32768	49152	65536	65537	81920	81921	86017	114688	147456	196608	196800	196992	212992	212993	213184	213376	245760
0	67	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	2
1	0	20	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
16384	0	0	210	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0
16385	0	0	0	21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
32768	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
49152	0	0	1	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65536	0	0	0	0	0	0	44	0	5	2	0	0	0	1	0	0	0	0	0	0	0
65537	0	2	0	0	1	0	0	3	1	0	0	0	0	0	1	0	0	0	0	0	0
81920	2	0	0	0	0	0	0	1	48	0	0	1	1	1	0	0	2	0	0	0	0
81921	0	0	0	0	0	0	2	0	0	3	0	0	0	0	0	0	0	1	0	0	0
86017	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
114688	1	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0
147456	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1
196608	4	0	0	0	0	0	3	0	2	0	0	0	0	79	0	0	0	0	0	0	1
196800	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
196992	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0
212992	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	152	0	0	0	1
212993	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0
213184	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0	0
213376	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
245760	1	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3

Figura C2. Matriz de Confusión del Modelo Gradient Boosting Classifier (GBC)

PREDICCIÓN

REAL	0	1	16384	16385	32768	49152	65536	65537	81920	81921	86017	114688	147456	196608	196800	196992	212992	212993	213184	213376	245760
0	65	0	0	0	0	0	0	0	2	0	0	0	0	12	0	0	0	0	0	0	0
1	0	22	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
16384	0	0	199	0	0	1	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0
16385	0	0	0	21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
32768	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0
49152	3	0	2	0	0	10	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
65536	0	0	0	0	0	0	47	0	1	0	0	0	0	4	0	0	0	0	0	0	0
65537	0	2	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
81920	2	0	0	0	0	0	0	0	49	0	0	2	1	0	0	0	2	0	0	0	0
81921	0	1	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	1	0	0	0
86017	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
114688	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1
147456	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0
196608	12	0	0	0	0	0	3	0	3	0	0	0	0	70	0	0	0	1	0	0	0
196800	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
196992	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
212992	0	0	3	0	0	1	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0
212993	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
213184	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	4	0	0
213376	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
245760	2	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	2

Figura C3. Matriz de Confusión del Modelo K-Nearest Neighbors (KNN)

PREDICCIÓN

REAL	0	1	16384	16385	32768	49152	81920	131072	131073	147456	147457	151553	180224	196608	212992	212993	213184	245760
0	73	0	0	1	0	1	0	0	0	4	0	0	0	6	0	0	0	1
1	0	12	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0
16384	0	0	212	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16385	3	0	0	24	0	0	0	0	0	0	0	0	0	4	0	0	0	1
32768	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
49152	1	0	1	0	0	14	0	0	0	0	0	0	0	0	0	0	0	2
81920	0	0	0	0	0	0	4	0	1	0	0	0	0	0	0	0	0	0
131072	0	0	0	0	0	0	0	40	0	2	1	0	0	0	0	1	0	0
131073	0	0	0	0	1	0	0	0	10	0	0	0	0	0	0	0	0	0
147456	2	1	0	0	0	0	0	1	0	46	0	0	0	2	1	0	0	0
147457	0	0	0	0	0	0	0	1	0	0	8	0	0	0	0	0	0	0
151553	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
180224	2	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0
196608	6	0	0	4	0	0	0	1	0	3	0	0	0	77	0	0	0	0
212992	0	0	1	0	0	0	0	0	0	1	0	0	0	0	147	0	0	1
212993	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
213184	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	3	0
245760	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	3

Figura C4. Matriz de Confusión del Modelo Extra Trees Classifier (ETC)

REAL \ PREDICCIÓN	0	1	16384	16385	32768	49152	65536	65537	81920	81921	86017	114688	147456	196608	196800	196992	212992	212993	213184	213376	245760
0	67	0	0	0	0	0	0	0	1	0	0	0	0	11	0	0	0	0	0	0	0
1	0	20	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
16384	0	0	206	0	0	1	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
16385	0	0	0	21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
32768	0	0	0	0	1	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0
49152	0	0	0	0	0	12	0	0	1	0	0	0	0	0	0	0	2	0	0	0	2
65536	0	0	0	0	0	0	49	1	0	0	0	0	0	2	0	0	0	0	0	0	0
65537	0	2	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
81920	3	0	0	0	0	0	1	0	47	0	0	0	1	1	0	0	3	0	0	0	0
81921	0	1	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
86017	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
114688	0	0	0	0	0	0	1	0	1	0	0	0	0	2	0	0	0	0	0	0	0
147456	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
196608	11	0	0	2	0	0	2	0	2	0	0	0	0	72	0	0	0	0	0	0	0
196800	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
196992	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
212992	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	155	0	0	0	0
212993	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
213184	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
213376	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
245760	1	0	0	1	0	1	0	0	2	0	0	0	0	1	0	0	1	0	0	0	1

Figura C5. Matriz de Confusión del Modelo Support Vector Classifier (SVC)

REAL \ PREDICCIÓN	0	1	16384	16385	32768	49152	65536	65537	81920	81921	86017	114688	147456	196608	196800	196992	212992	212993	213184	213376	245760
0	66	0	0	0	0	1	0	0	1	0	0	0	0	11	0	0	0	0	0	0	0
1	0	20	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
16384	0	0	209	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
16385	0	0	0	21	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
32768	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0
49152	1	0	1	0	0	14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
65536	0	0	0	0	0	0	50	0	1	0	0	0	0	1	0	0	0	0	0	0	0
65537	0	3	0	0	0	0	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0
81920	1	0	0	0	0	0	1	0	48	0	0	0	1	3	0	0	2	0	0	0	0
81921	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
86017	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
114688	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0
147456	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
196608	2	0	0	0	0	0	2	0	0	0	0	0	0	85	0	0	0	0	0	0	0
196800	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
196992	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0
212992	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	154	0	0	0	0
212993	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0
213184	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
213376	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
245760	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

Figura C6. Matriz de Confusión del Modelo Random Forest Classifier (RFC)