

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Creación de modelo de monitoreo no intrusivo de cargas (NILM) para
desagregación eléctrica aplicado en Ecuador.

PROYECTO DE TITULACIÓN

Previo la obtención del Título de:

Magister en Ciencias de Datos

Presentado por:

Carlos Enrique Cedeño Zamora

GUAYAQUIL - ECUADOR

Año: 2024

DEDICATORIA

A mis dos madres,
Cecilia y Mariana.
Todo lo que soy es gracias a Uds.

AGRADECIMIENTOS

Un sincero agradecimiento al PhD. José Córdova, quien ha sido un gran tutor y amigo desde mis épocas de pre-grado.

De igual forma al PhD. Miguel Torres, quien me brindo asesoría en el campo eléctrico para este trabajo.

Por último, pero no menos importante a mis grandes amigos, los Ings. Vaccaro y Cruz, y las Ings. Coello y Riquero por su colaboración oportuna en la implementación de este proyecto.

COMITÉ EVALUADOR

PhD. José Córdova
Tutor

MSc. Allan Avendaño
Revisor

RESUMEN

El uso de técnicas de monitoreo en el campo eléctrico ha revolucionado la toma de decisiones para los administradores de edificios y sistemas de control de energía. Además, la creciente disponibilidad de sensores y smart-meters permite la creación de aplicaciones en entornos inteligentes, como la identificación de patrones de ocupación en oficinas y edificios. Sin embargo, la falta de datasets abiertos para analizar y probar tales aplicaciones retrasa su implementación y complica la adaptación a escenarios desconocidos. En este trabajo, se presenta el proceso de creación de un modelo para la desagregación no intrusiva de cargas eléctricas o NILM, el cual consiste en obtener las mediciones desagregadas de una fuente primaria de monitoreo eléctrico sin necesidad de sensores adicionales. Esto implica varias fases, desde la creación de un sistema de telemetría para obtener los datos, el armado de un dataset, creación de perfiles de carga, el entrenamiento de modelos y la puesta en producción, todo esto presentado en un dashboard de monitoreo. Para aprovechar la fuente de datos que se va a crear, se complementó la información con estadísticas en tiempo real de la operación eléctrica tales como el consumo acumulado, pronósticos de consumo, su equivalente monetario y otros parámetros de importancia.

Al final, presentamos CLED, *Computer Lab Level Energy Dataset*, un conjunto de datos obtenido en un entorno controlado con dispositivos comúnmente encontrados en un laboratorio de aula de ingeniería. También NilmEC, una aplicación web que permite el monitoreo en tiempo real de parámetros de operación de una red eléctrica, con pronósticos de consumo y facturación, datos históricos y desagregación de cargas no intrusiva.

Palabras clave: NILM, monitoreo eléctrico, dataset de energía, telemetría

ABSTRACT

The use of monitoring techniques in the electrical field has revolutionized decision-making for building administrators and energy control systems. Additionally, the increasing availability of sensors and smart meters enables the creation of applications in smart environments, such as the identification of occupancy patterns in offices and buildings. However, the lack of open datasets for analyzing and testing such applications delays their implementation and complicates adaptation to unknown scenarios. This work presents the process of creating a model for non-intrusive load monitoring (NILM), which involves obtaining disaggregated measurements from a primary source of electrical monitoring without the need for additional sensors. This process involves several phases, from creating a telemetry system to obtain the data, building a dataset, creating load profiles, training models, and deploying them, all presented in a monitoring dashboard. To leverage the created data source, real-time statistics of electrical operation such as cumulative consumption, consumption forecasts, their monetary equivalent, and other important parameters were included. Finally, we present CLED, the Computer Lab Level Energy Dataset, a dataset obtained in a controlled environment with devices commonly found in an engineering classroom laboratory. We also introduce NilmEC, a web application that allows real-time monitoring of electrical network operation parameters, with consumption forecasts and billing, historical data, and non-intrusive load disaggregation.

Keywords: NILM, energy monitoring, energy dataset, telemetry

ÍNDICE GENERAL

| | |
|-------------------------------------|-----|
| Dedicatoria..... | I |
| Agradecimientos | II |
| comité evaluador..... | III |
| Resumen | IV |
| Abstract..... | V |
| ÍNDICE GENERAL..... | VI |
| Abreviaturas..... | IX |
| índice de figuras..... | X |
| índice de Tablas..... | XII |
| CAPÍTULO 1 | 14 |
| 1. Planteamiento del problema | 14 |
| 1.1 Descripción del problema | 14 |
| 1.2 Justificación del problema..... | 15 |
| 1.3 Solución propuesta | 15 |
| 1.4 Objetivos..... | 16 |
| 1.4.1 Objetivo General | 16 |
| 1.4.2 Objetivos Específicos | 16 |
| 1.5 Metodología | 17 |
| 1.6 Resultados esperados | 17 |
| 1.7 Dataset | 19 |
| CAPÍTULO 2..... | 21 |
| 2. EStado del arte | 21 |
| 2.1 Marco referencial | 21 |
| 2.1.1 Teoría de NILM | 22 |

| | | |
|-----------------|--|----|
| 2.1.2 | Caracterización de dispositivos para desagregación | 23 |
| 2.2 | Arquitectura común usada para desagregación | 24 |
| 2.3 | NILM datasets..... | 25 |
| 2.4 | Algoritmos para desagregación | 26 |
| 2.4.1 | Mean | 27 |
| 2.4.2 | Edge Detection..... | 27 |
| 2.4.3 | Combinatorial Optimization | 28 |
| 2.4.4 | Discriminative Sparse Coding | 28 |
| 2.4.5 | Exact FHMM, Aproximate FHMM | 28 |
| 2.4.6 | Denoising autoencoder | 28 |
| 2.4.7 | RNN | 29 |
| 2.4.8 | Seq2Point & Seq2Seq..... | 29 |
| 2.5 | Pronóstico de consumo energético..... | 29 |
| CAPÍTULO 3..... | | 32 |
| 3. | Diseño e Implementación..... | 32 |
| 3.1 | Diseño de sistema de telemetría | 32 |
| 3.1.1 | Conexiones físicas para medidores de energía IoT | 32 |
| 3.1.2 | Configuración lógica para medidores de energía IoT | 34 |
| 3.2 | Creación de dashboard de monitoreo..... | 37 |
| 3.2.1 | Historias de usuario..... | 38 |
| 3.2.2 | Definición de módulos del dashboard | 39 |
| 3.3 | Perfil de carga de dispositivos | 40 |
| 3.3.1 | Método visual – Histograma..... | 41 |
| 3.3.2 | K-means clustering..... | 43 |
| 3.3.1 | Gaussian Mixture Models para clustering | 44 |
| 3.3.2 | Estimaciones con Kernel Density | 45 |

| | | |
|-----------------|--|-----|
| 3.4 | Desarrollo de modelo de pronóstico de consumo y facturación de energía eléctrica | 46 |
| 3.5 | Desarrollo de modelo de desagregación no intrusivo de cargas | 52 |
| CAPÍTULO 4..... | | 57 |
| 4. | Análisis de resultados | 57 |
| 4.1 | Dataset para desagregación de cargas | 57 |
| 4.1.1 | Creación del dataset en formato HDF5 para NILMTK..... | 58 |
| 4.2 | Estados de operación por dispositivo | 65 |
| 4.2.1 | Proyector..... | 66 |
| 4.2.2 | Monitor | 68 |
| 4.2.3 | Router | 70 |
| 4.2.4 | Switch..... | 71 |
| 4.2.5 | PC's..... | 72 |
| 4.2.6 | Aire acondicionado..... | 77 |
| 4.3 | Modelo NILM | 80 |
| 4.4 | Modelo de pronóstico para la facturación de consumo eléctrico | 85 |
| 4.5 | Dashboard de monitoreo de consumo energético y desagregación de cargas. 89 | |
| 4.5.1 | NilmEC | 90 |
| 4.5.2 | NilmEC-API y NilmEC-NILM | 96 |
| CAPÍTULO 5..... | | 99 |
| 5. | Conclusiones y recomendaciones..... | 99 |
| 6. | Bibliografía | 102 |

ABREVIATURAS

| | |
|-------|---|
| ESPOL | Escuela Superior Politécnica del Litoral |
| NILM | Non-Intrusive Load Monitoring |
| IOT | Internet of Things |
| REDD | Reference Energy Disaggregation dataset |
| CLED | Computer lab energy dataset |
| FSM | Finite state machines |
| CVD | Continuous variable devices |
| HES | Household Electricity Survey |
| MEULP | Midwest Energy Usage Load Profiles |
| REFIT | Refit Smart Home dataset |
| AMPDS | Almanac of Minutely Power Dataset |
| ECO | Electricity Consumption and Occupancy |
| DRED | Dutch Residential Energy Dataset |
| CO | Combinatorial Optimization |
| FHMM | Factorial Hidden Markov Model |
| RNN | Recurrent neural networks |
| LSTM | Long Short Term Memory |
| MQTT | Message queuing telemetry transport |
| W | Watts |
| kWh | kilo-vatios hora |
| GMM | Gaussian Mixture Models |
| PDF | Probability density function |
| KDE | Kernel Density |
| CSV | Comma separated values |
| LST | Laboratorio de Sistemas Telemáticos |
| IEEE | Institute of Electrical and Electronics Engineers |
| HDF5 | Hierarchical Data Format version 5 |
| AP | Access Point |
| BTU | British Thermal Units |
| YAML | YAML Ain't Markup Language |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Ilustración 1 - Resultado esperado #1 - Creación de dataset de dispositivos..... | 17 |
| Ilustración 2 - Resultado esperado #2 - Análisis de patrones de consumo para creación de un perfil de carga de dispositivos..... | 18 |
| Ilustración 3 - Resultado esperado #3 - Modelo de desagregación eléctrica de cargas no intrusiva..... | 18 |
| Ilustración 4 - Resultado esperado #4 - Modelo de pronóstico de consumo eléctrico | 19 |
| Ilustración 5 - Resultado esperado #5 - Modelo de telemetría para medidores inteligentes..... | 19 |
| Ilustración 6. Consumo total de energía versus tiempo mostrando el aporte individual de cada dispositivo (HART, 1992) | 22 |
| Ilustración 7. Diagrama de bloques generalizada de las arquitecturas para desagregación orientada a datos (Schirmer & Mporas, 2023)..... | 25 |
| Ilustración 8. Pipeline de NILMTK. En cada paso del sistema los datos pueden ser almacenados en memoria o disco (Batra & Kelly, 2014) | 27 |
| Ilustración 9. Modelo de EPC-PM para pronóstico de consumo energético (Kumar & Gupta, 2023)..... | 30 |
| Ilustración 10 Esquema de conexión para el dispositivo Shelly 3EM..... | 33 |
| Ilustración 11 Imagen referencial del dispositivo Sonoff S31 | 33 |
| Ilustración 12 Imagen referencial del proceso de comunicación serial-USB para carga de firmware | 34 |
| Ilustración 13 Diagrama de arquitectura para telemetría | 36 |
| Ilustración 14 Diagrama de conexión MQTT de alto nivel | 37 |
| Ilustración 15 Vista general del dashboard | 40 |
| Ilustración 16 Modo suspensión - Ahorro de energía..... | 42 |
| Ilustración 17 Histograma de la Potencia activa. Bins=80 | 42 |
| Ilustración 18. Método de inercia para encontrar K óptima | 44 |
| Ilustración 19 Ajuste de curva mediante el método KDE | 46 |
| Ilustración 20. Conexiones de fase para el medidor Shelly 3EM | 48 |

| | |
|---|----|
| Ilustración 21. Fragmento tomado del Pliego Tarifario del Servicio Público de Energía Eléctrica 2024 | 50 |
| Ilustración 22. Representación a alto nivel de Chronos (Ansari, 2024)..... | 51 |
| Ilustración 23 Diagrama de procesos para la generación del modelo de desagregación de cargas eléctricas | 56 |
| Ilustración 24. Inercia para el dispositivo proyector | 67 |
| Ilustración 25. Inercia para dispositivo Monitor | 69 |
| Ilustración 26. Gráfico de frecuencias para el dispositivo Router | 71 |
| Ilustración 27. Gráfico de frecuencias para dispositivo Switch | 72 |
| Ilustración 28. Gráficos de frecuencia para PC0 a PC4..... | 73 |
| Ilustración 29. Grafico de frecuencias para el Aire Acondicionado | 78 |
| Ilustración 30. Representación gráfica de las señales de los dispositivos y medidores con metadata | 82 |
| Ilustración 31. Representación de las señales eléctricas sin clasificar por metadata | 83 |
| Ilustración 32. Periodos de actividad/inactividad de los dispositivos eléctricos | 84 |
| Ilustración 33. Predicciones de consumo, variación de acuerdo al tamaño del contexto, Diciembre 2023..... | 86 |
| Ilustración 34. Predicciones de consumo, variación de acuerdo al tamaño del contexto, Junio 2024 | 88 |
| Ilustración 35. NilmEC primera sección | 90 |
| Ilustración 36. NilmEC segunda sección..... | 90 |
| Ilustración 37. Tarjetas de monitoreo: Consumo actual en kW y Gasto actual en USD | 91 |
| Ilustración 38. Tarjetas de monitoreo: Gasto mensual proyectado y Consumos pasados | 92 |
| Ilustración 39. Monitoreo en tiempo real, consumo trifásico | 93 |
| Ilustración 40. Opciones de exportado de datos, anotación e impresión | 93 |
| Ilustración 41. Parámetros de operación de la red eléctrica en tiempo real..... | 94 |
| Ilustración 42. Gráfica de desagregación eléctrica no intrusiva. | 94 |
| Ilustración 43. Selección de día para la desagregación eléctrica..... | 95 |
| Ilustración 44. Gráfica de porción de energía consumida por dispositivo | 95 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Variables disponibles para medición en el equipo Shelly 3EM (Shelly, 2023) | 20 |
| Tabla 2. Variables disponibles para medición en el equipo Sonoff S31 | 20 |
| Tabla 3. Comparación de características de varios datasets NILM | 26 |
| Tabla 4. Formato de datos de cada dispositivo IoT en el estudio | 37 |
| Tabla 5. Descripción de Hito 1 | 38 |
| Tabla 6. Descripción de Hito 2 | 38 |
| Tabla 7. Descripción de Hito 3 | 38 |
| Tabla 8. Descripción de Hito 4 | 38 |
| Tabla 9. Descripción de Hito 5 | 39 |
| Tabla 10. Descripción de Hito 6 | 39 |
| Tabla 11. Descripción de Hito 7 | 39 |
| Tabla 12. Descripción de Hito 8 | 39 |
| Tabla 13. Resultados de clustering para $k=3$ y $K=4$ | 44 |
| Tabla 14. Resultados estadísticos para $k=3$ y $K=4$ | 45 |
| Tabla 15. Resultados de método KDE para obtención de clústeres | 46 |
| Tabla 16. Lista de dispositivos incluidos en el estudio | 57 |
| Tabla 17. Proyector Kmeans | 66 |
| Tabla 18. Proyector GMM | 66 |
| Tabla 19. Proyector KDE | 67 |
| Tabla 20. Estados de operación del Proyector | 68 |
| Tabla 21. Monitor Kmeans | 68 |
| Tabla 22. Monitor GMM | 69 |
| Tabla 23. Monitor KDE | 69 |
| Tabla 24. Estados de operacion del Monitor | 70 |
| Tabla 25. Modelos de Router comunes | 70 |
| Tabla 26. Estados de operacion del router | 71 |
| Tabla 27. Estados de operación del switch | 72 |
| Tabla 28. PC0 Kmeans | 74 |

| | |
|--|----|
| Tabla 29. PC0 GMM | 74 |
| Tabla 30. PC0 KDE..... | 74 |
| Tabla 31. PC1 Kmeans..... | 75 |
| Tabla 32. PC1 GMM | 75 |
| Tabla 33. PC1 KDE..... | 75 |
| Tabla 34. PC2 Kmeans..... | 75 |
| Tabla 35. PC2 GMM | 75 |
| Tabla 36. PC2 KDE..... | 76 |
| Tabla 37. PC3 Kmeans..... | 76 |
| Tabla 38. PC3 GMM | 76 |
| Tabla 39. PC3 KDE..... | 76 |
| Tabla 40. PC4 Kmeans..... | 76 |
| Tabla 41. PC4 GMM | 77 |
| Tabla 42. PC4 KDE..... | 77 |
| Tabla 43. Estados de operación de las PC..... | 77 |
| Tabla 44. AC Kmeans..... | 78 |
| Tabla 45. AC GMM | 79 |
| Tabla 46. AC KDE..... | 79 |
| Tabla 47. Métricas de evaluación para los algoritmos de desagregación eléctrica.... | 84 |
| Tabla 48. Métricas de error para la variación del tamaño de contexto del mes Diciembre 2023..... | 87 |
| Tabla 49. Métricas de error para la variación del tamaño de contexto del mes Junio 2023..... | 88 |
| Tabla 50. API Endpoints del backend NilmEC-API | 96 |
| Tabla 51. Endpoint del backend NilmEC-NILM..... | 98 |

CAPÍTULO 1

1. PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción del problema

NILM, o Non-Intrusive Load Monitoring (Monitorización de Carga No Intrusiva), es una técnica que se utiliza para rastrear y analizar el consumo de energía eléctrica en un hogar o un edificio sin la necesidad de instalar sensores en cada dispositivo eléctrico. En lugar de eso, NILM utiliza algoritmos y técnicas de procesamiento de señales para descomponer el consumo total de energía en diferentes cargas individuales.

Al ser un método no intrusivo, facilita la implementación y evita costos adicionales y molestias asociados con la instalación de hardware.

Desde la década de los 80s ha existido el esfuerzo de aplicar NILM para identificar patrones de uso, consumo y sobre todo anomalías causadas por deficiencia en la operación de equipos que consumen electricidad.

Durante años este problema tuvo soluciones en el campo de la optimización, hasta llegar a nuestros días donde los esfuerzos en ML han proporcionado algunas metodologías y frameworks, creando modelos que pueden ser usados en dispositivos que cumplan el objetivo de NILM.

Sin embargo, estos modelos y equipos han sido entrenados usando estándares americanos o europeos que difieren significativamente con la realidad ecuatoriana. Para entrenar un modelo NILM, primero se debe recopilar datos de consumo eléctrico de una o más viviendas o edificios. Estos datos deben incluir medidas de consumo eléctrico tomadas en intervalos regulares, como lecturas de medidores de energía, datos de sensores de corriente y voltaje, e incluso analizadores especializados de energía.

Los distintos estándares de energía que cada país tiene representan un fuerte sesgo de entrenamiento cuando hablamos de este tipo de problemas. El sesgo ocurre cuando los datos utilizados para entrenar el modelo no son representativos de la diversidad de situaciones que el modelo puede encontrar en el mundo real. Esto puede conducir a que

el modelo sea menos preciso al desagregar cargas eléctricas en entornos que difieren significativamente de los datos de entrenamiento.

Esto limita a investigadores e ingenieros a estudiar este problema de forma teórica sin un uso real en el Ecuador.

1.2 Justificación del problema

Cuando se intentan aplicar los modelos de NILM tradicionales a un escenario ecuatoriano estos no tienen el mismo rendimiento que el reportado por investigadores alrededor del mundo. Lo cual despierta la necesidad de tener un dataset y modelo de desagregación de energía que aplique a la realidad del país.

La nueva era digital acompañada del auge de dispositivos IOT ha hecho que se puedan tener disponibles equipos como sensores y medidores “inteligentes” de agua, luz, gas, etc. Esto permite que investigadores puedan recolectar data de forma mucho más amigable y sencilla que hasta hace unos pocos años. Existen datasets para desagregación de energía como REDD-2011 los cuales en su desarrollo tuvieron que construir hardware especializado con un alto costo de implementación y complejidad de instalación y mantenimiento. Actualmente hay disponibles en el mercado dispositivos que rondan los cientos de dólares y cuentan con las mismas capacidades de data logging y ofrecen mejores alternativas de conectividad (wifi, bluetooth) lo que simplifica la instalación.

La disponibilidad de smartmeters, combinados con un diseño de sistema de recolección de datos distribuidos para sensores en un ambiente controlado permitirán recolectar la data necesaria para crear un perfil de los principales equipos electrónicos en el país y de esta forma crear modelos que puedan ser aplicados en el Ecuador.

1.3 Solución propuesta

Para abordar el problema expuesto se debe empezar a recopilando datos de dispositivos en operación en la red eléctrica ecuatoriana, para luego caracterizarlos, generar las desagregaciones y visualizar los resultados de una forma amigable y representativa.

Se resume la propuesta en 5 ejes principales:

- a. Diseño de un sistema de telemetría para la recopilación de data de dispositivos.
- b. Armado de dataset
- c. Entrenamiento y validación de modelo
- d. Creación de pipeline para aplicar un método de optimización
- e. Creación de un dashboard par visualización en tiempo real de estadísticas de interés.

1.4 Objetivos

1.4.1 Objetivo General

Generar un modelo de NILM preciso para la desagregación de cargas eléctricas utilizando técnicas de aprendizaje automático.

1.4.2 Objetivos Específicos

- Recopilar y preprocesar datos de consumo de energía eléctrica, incluyendo datos de demanda total de energía eléctrica y datos de consumo de equipos individuales.
- Evaluar la precisión y escalabilidad de diferentes modelos de aprendizaje automático para la desagregación de cargas eléctricas.
- Optimizar los modelos seleccionados para lograr el mejor rendimiento en términos de precisión y escalabilidad, mediante la experimentación con diferentes parámetros y técnicas de preprocesamiento de datos.
- Validar el modelo final a través de pruebas de datos de consumo de energía eléctrica reales, para demostrar su eficacia en la desagregación de cargas eléctricas.

1.5 Metodología

- Diseñar e implementar un modelo de telemetría basado en nube privada usando proyectos de código abierto.
- Recolectar datos para caracterizar diversos dispositivos en el estudio.
- Usar bases de datos de series temporales para almacenar, explorar y servir datos.
- Realizar un preprocesamiento de los datos, dimensionando estados de operación de los dispositivos objetos de estudio.
- Probar distintos métodos del Framework NEURALNILM – NILMTK para comprobar nuestra solución.
- Crear un pipeline de datos donde el resultado de la desagregación pasa por un proceso de optimización.

1.6 Resultados esperados

- Creación de un dataset de consumo eléctrico para dispositivos comunes.

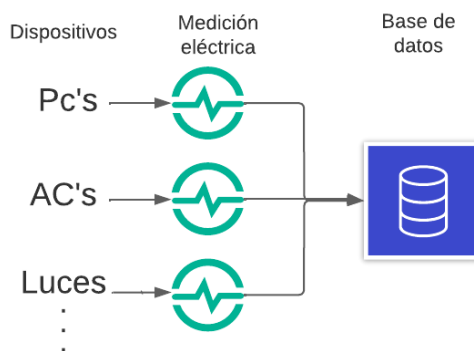


Ilustración 1 - Resultado esperado #1 - Creación de dataset de dispositivos

- Creación de un perfil de cargas de los dispositivos usados en el estudio, caracterizando modos de operación y potencias nominales.

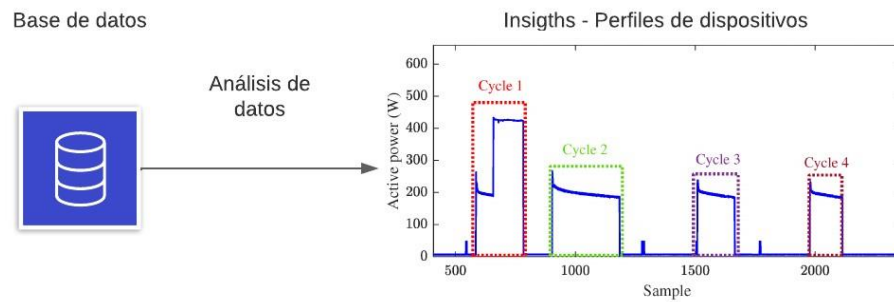


Ilustración 2 - Resultado esperado #2 - Análisis de patrones de consumo para creación de un perfil de carga de dispositivos

- Modelo de NILM, usando los principales modelos de redes neuronales presentes en el framework NILMTK.

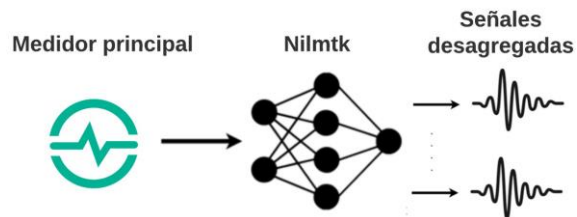


Ilustración 3 - Resultado esperado #3 - Modelo de desagregación eléctrica de cargas no intrusiva

- Modelo de pronósticos de facturación y consumo energético considerando los consumos históricos y actuales.

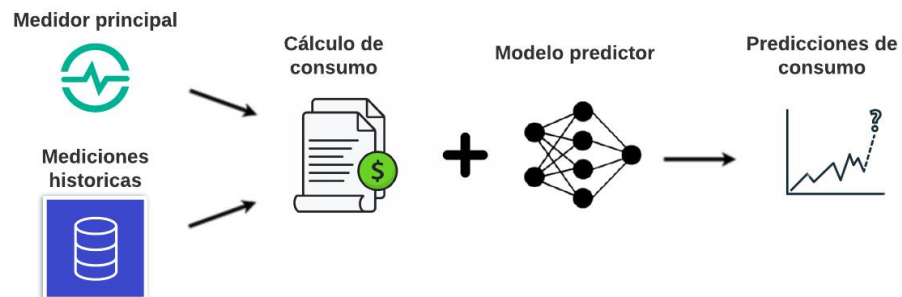


Ilustración 4 - Resultado esperado #4 - Modelo de pronóstico de consumo eléctrico

- Diseño del sistema de telemetría, con un dashboard de monitoreo de consumo y desagregación.

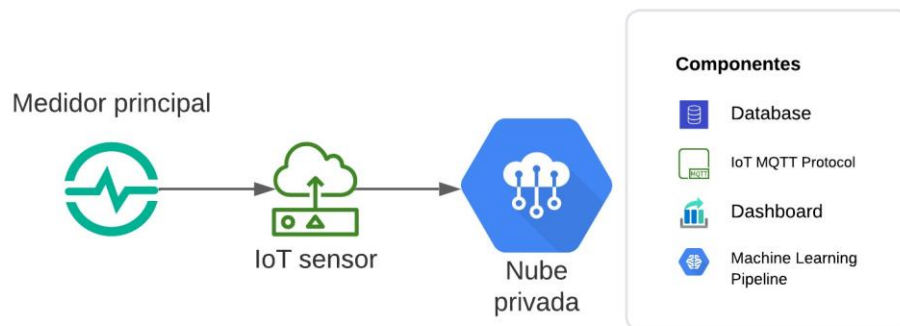


Ilustración 5 - Resultado esperado #5 - Modelo de telemetría para medidores inteligentes

1.7 Dataset

Uno de los puntos principales de este trabajo es recopilar data energética de distintos dispositivos para la creación de un dataset que represente las cargas que se encontrarían en un Campus Universitario, tomando como referencia el Laboratorio de Sistemas Telemáticos de ESPOL.

Esta creación de dataset comienza con un monitoreo intrusivo mediante el uso de medidores de energía individuales para los dispositivos eléctricos seleccionados para su caracterización y el uso de un medidor de energía trifásico para obtener la lectura general del panel de distribución. A continuación, se detallan las variables disponibles.

Tabla 1. Variables disponibles para medición en el equipo Shelly 3EM (Shelly, 2023)

| Nombre del equipo: Shelly 3EM | |
|--|---------------|
| Ubicación: Laboratorio de Sistemas Telemáticos – Panel de distribución | |
| Variable | Unidad |
| Corriente (por fase) | Amperios |
| Voltaje (por fase) | Volts |
| Energía (por fase) | kWh |
| Potencia (por fase) | Watts |
| Factor de potencia (por fase) | - |

Tabla 2. Variables disponibles para medición en el equipo Sonoff S31

| Nombre del equipo: Sonoff S31 | |
|---|---------------|
| Ubicación: Laboratorio de Sistemas Telemáticos – por equipo | |
| Variable | Unidad |
| Corriente (monofásica) | Amperios |
| Voltaje (monofásica) | Volts |
| Potencia Aparente | VA |
| Potencia Reactiva | VAR |
| Potencia Activa | Watts |
| Factor de potencia | - |

Ambos dispositivos ingestan datos a una frecuencia de muestreo de 10 segundos en una base de datos para series temporales, cuyo diseño se explicará en capítulos posteriores.

CAPÍTULO 2

2. ESTADO DEL ARTE

En este capítulo se realiza una recopilación y análisis de artículos científicos y trabajos previos que muestran el proceso de realizar la desagregación no intrusiva de cargas, los principales métodos y conjuntos de datos. En la otra arista, analizaremos algunos métodos para realizar la predicción del consumo energético sobre un periodo definido.

2.1 Marco referencial

El concepto de NILM se originó en la década de 1980, cuando los investigadores comenzaron a explorar formas de desagregar el consumo de energía eléctrica en cargas individuales sin la necesidad de sensores intrusivos. Los primeros enfoques se basaron en técnicas analógicas y de procesamiento de señales. En la década de 1990 (HART, 1992), con el crecimiento de las redes eléctricas inteligentes y la disponibilidad de datos de consumo más detallados, se produjo un avance significativo en NILM. Se aplicaron técnicas de procesamiento de señales digitales y métodos estadísticos para mejorar la precisión y la eficiencia de la desagregación de cargas.

A medida que los algoritmos de aprendizaje automático y la inteligencia artificial comenzaron a ganar protagonismo, se aplicaron enfoques como redes neuronales, árboles de decisión y algoritmos de agrupamiento para mejorar aún más la capacidad de desagregación de NILM. Estos avances permitieron una mayor precisión en la identificación de dispositivos y una mejor adaptabilidad a diferentes entornos.

En los últimos años, NILM ha pasado de ser una tecnología principalmente utilizada en entornos de investigación a ser implementada en aplicaciones prácticas. Se han desarrollado sistemas comerciales de NILM para el monitoreo de energía en hogares y edificios, lo que ha permitido a los usuarios comprender mejor su consumo eléctrico y tomar medidas para la eficiencia energética.

Aunque NILM ha experimentado avances significativos, todavía existen desafíos en su implementación. Algunos de estos desafíos incluyen la variabilidad en los patrones de consumo, la interferencia electromagnética, sesgo en los datos, falta de datasets completos para distintos escenarios y la necesidad de conjuntos de datos etiquetados para el entrenamiento de algoritmos.

2.1.1 Teoría de NILM

En el primer acercamiento formal hacia la definición de concepto y bases para la monitorización no intrusiva (HART, 1992) define que el objetivo principal es determinar el consumo de energía que producen los distintos dispositivos conectados a la red eléctrica.

Mediante un análisis de las formas de onda de voltaje y corriente de la carga total del sistema, podemos estimar el número y naturaleza de las cargas individuales. Esto proporciona un método muy conveniente y efectivo para recopilar datos de carga en comparación con los medios tradicionales de colocar sensores en cada uno de los componentes individuales de la carga.

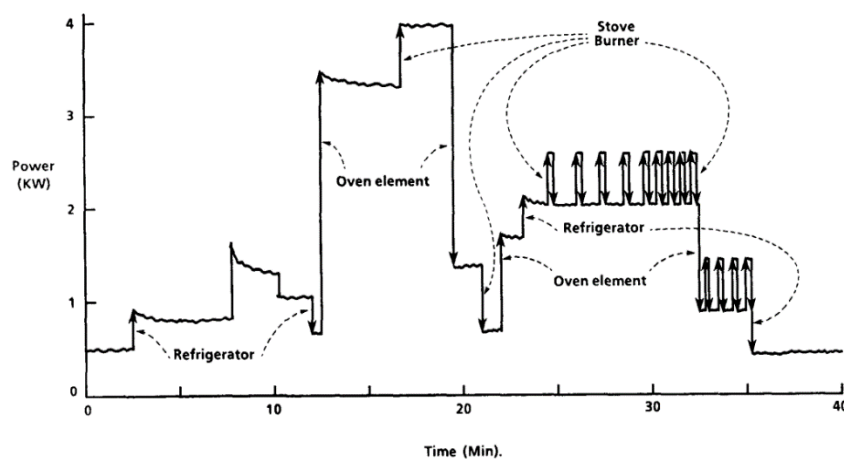


Ilustración 6. Consumo total de energía versus tiempo mostrando el aporte individual de cada dispositivo (HART, 1992)

Para desagregar la carga global en sus elementos constituyentes, se emplea un enfoque basado en modelos para describir los dispositivos individuales y su interacción. Estos modelos sugieren ciertas características

distintivas que pueden identificarse en la carga total para señalar los estados de los componentes por separado.

Para simplificar esta sección asumamos que existen solo dos tipos de estados (on / off). Suponiendo que existen n dispositivos y que la potencia total consumida en el tiempo t es la suma de todas las señales de potencia individuales $P_i(t)$.

Para modular el proceso de switch de los dispositivos tenemos la función $s[t]$ donde:

$$s(t) = \begin{cases} 1, & \text{Si el dispositivo en el tiempo } t \text{ esta encendido} \\ 0, & \text{Si el dispositivo en el tiempo } t \text{ esta apagado} \end{cases} \quad (1)$$

La potencia total puede ser estimada mediante la siguiente formula:

$$P_{total}(t) = \sum_{i=1}^n s_i(t)P_i(t) + e(t) \quad (2)$$

Donde el termino aditivo $e(t)$ representa el error de la modelización de los dispositivos individualmente. El término de error puede ser indetectable en dispositivos continuamente encendidos, o irrastable en dispositivos que varían continuamente o tienen múltiples estados. El objetivo general es minimizar el término de error (Klemenjak & Goldsborough, 2016).

2.1.2 Caracterización de dispositivos para desagregación

El término “dispositivo” mencionado en las secciones anteriores hace referencia a cualquier carga de la red eléctrica cuyo consumo puede ser identificado en el medidor principal de la red. La identificación de estos dispositivos depende fuertemente de su comportamiento, lo cual nos obliga a caracterizarlos en cuatro distintas categorías (Zoha & Gluhak, 2012):

- i. Tipo-1: Dispositivos que tienen solo dos estados de operación (ON/OFF). Un ejemplo claro de este tipo son los focos que no posean un dimmer.
- ii. Tipo-2: Dispositivos que tienen un número finito de estados (multi-estados). Estos también son referidos como *Máquinas de Estado Finitos* (FSM), y poseen varias formas de operación, lo cual impacta directamente en el consumo del dispositivo, por ejemplo, una lavadora que dependiendo del modo de operación en que trabaje, puede prender o no bombas o tornos que se traduce en un mayor consumo energético.
- iii. Tipo-3: Dispositivos conocidos como *Continuamente Variables* (CVD) , donde su consumo energético no posee un patrón repetitivo o fijo, sino que varía entre un rango conocido dependiendo de variables de operación. Un ejemplo de esto pueden ser las luces con dimmer.
- iv. Tipo-4: Dispositivos que usualmente permanecen encendidos consumiendo una cantidad constante y usualmente pequeña de energía. Este consumo puede ser considerado como un offset del sistema. Ejemplo de estos podrían ser detectores de humo, routers inalámbricos, teléfonos digitales.

2.2 Arquitectura común usada para desagregación

Dependiendo de la complejidad del escenario, no todas las arquitecturas usadas en la literatura comparten los mismos pasos como pre o post procesamiento, pero usualmente siguen el proceso mostrado en la siguiente figura (Schirmer & Mporas, 2023):

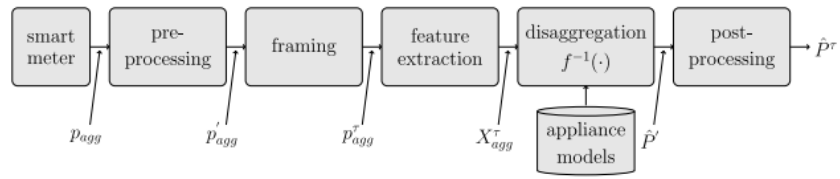


Ilustración 7. Diagrama de bloques generalizada de las arquitecturas para desagregación orientada a datos (Schirmer & Mporas, 2023)

Estos pasos consisten en un medidor Smart del que obtenemos la señal agregada P_{agg} , la cual pasa por un preprocesamiento P'_{agg} , y se aplica una ventana de tiempo fija P^{τ}_{agg} para obtener las propiedades de la señal, resultando en una variable X^{τ}_{agg} , a la cual se aplica el método de extracción y un posible post-procesamiento, dando como resultado la estimación de los consumos individuales de cada dispositivo a partir de la señal agregada $\hat{P}^{\tau} = \{\hat{P}^{\tau}_1 + \hat{P}^{\tau}_2, \dots, \hat{P}^{\tau}_{N-1} + \hat{e}^{\tau}_1\}$

2.3 NILM datasets

Los conjuntos de datos de consumo de energía son de suma importancia para la investigación de NILM, ya que permiten desarrollar y validar modelos de aprendizaje automático y comprobar su rendimiento, siendo un elemento vital para reproducir los resultados experimentales obtenidos (Himeur & Alsalemi, 2022).

Estos datasets usualmente incorporan perfiles de consumo del circuito agregado de la fuente principal de energía y de los dispositivos estudiados, lo cual se conoce como *ground-data*. Las variables que usualmente suelen incluir estos conjuntos de datos son:

- I. Periodos de tiempo de recolección de datos
- II. Número de casas/edificios monitoreados
- III. Numero de dispositivos
- IV. Características eléctricas:
 - a. Voltaje
 - b. Potencia Activa
 - c. Potencia Reactiva
 - d. Potencia Aparente

- e. Potencia normalizada
- f. Energía
- g. Frecuencia de muestreo
- h. Angulo de fase
- i. Factor de potencia

Todas estas variables entre conjuntos de datos dependiendo el contexto y profundidad del estudio donde fueron realizadas. La siguiente tabla muestra una comparación entre algunos de los principales datasets disponibles (Alsalemi & Himeur, 2022)

Tabla 3. Comparación de características de varios datasets NILM

| Dataset | Periodo de muestreo (días) | # Dispositivos | Variables | Frecuencia de muestreo (Hz) |
|---------|----------------------------|----------------|------------------------|-----------------------------|
| REDD | 119 | 24 | I, V, P | 1/3 |
| HES | 365 | 23 | I, V, P, T | 1/600 |
| MEULPv1 | 365 | 8 | P | 1/60 |
| REFIT | 213 | 9 | P, pf, L | 1/8 |
| AMPds1 | 365 | 21 | I, V, P, Q, S, pf, f | 1/60 |
| ECO | 240 | 14 | I, V, ϕ | 1 |
| AMPds2 | 730 | 21 | I, V, P, S, F, pf | 1/60 |
| UK-DALE | 655 | 53 | P | 1/6 |
| DRED | 180 | 13 | P, T, H, Pr | 1/8 |

2.4 Algoritmos para desagregación

La falta de disponibilidad de implementaciones de código abierto de los algoritmos para NILM, hizo que muchos experimentos fueran difíciles de replicar. Los investigadores se vieron obligados a utilizar varios algoritmos estándar porque no había versiones disponibles de las técnicas state-of-the-art. Además, los algoritmos suelen usar diferentes métricas y conjuntos de datos, lo que dificulta la comparación justa entre ellos. Todo esto complicó la tarea de que un investigador nuevo determinara cuál era el algoritmo más efectivo.

Para abordar este problema, en 2014 se desarrolla el framework NILMTK (Batra & Kelly, 2014) con el objetivo de motivar a los investigadores a usar un conjunto de herramientas para desarrollar nuevos algoritmos y probarlos bajo una serie de escenarios unificados, facilitando así la replicabilidad de resultados. La siguiente figura muestra el flujo de datos desde la importación de datasets, ya sea datasets conocidos o un nuevo conjunto de datos, hasta la evaluación de distintas métricas. Se presenta el formato de datos NILMTK-DF, el cual se inspira en el formato usado en el conjunto de datos REDD. (Kolter & Johnson, 2011)

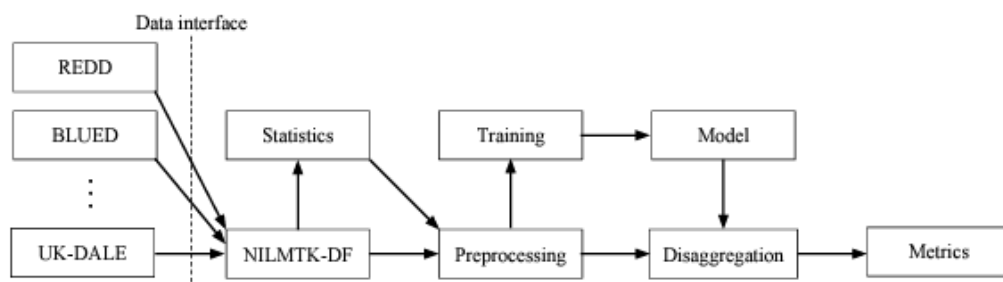


Ilustración 8. Pipeline de NILMTK. En cada paso del sistema los datos pueden ser almacenados en memoria o disco (Batra & Kelly, 2014)

Como parte de la expansión del proyecto principal, (Kukunuri & Batra, 2020) hace una extensión al proyecto principal con NILMTK-CONTRIB donde se presenta un API mejorado para usar los distintos algoritmos disponibles.

2.4.1 Mean

El algoritmo Mean (Promedio) estima el consumo de energía de un electrodoméstico como el promedio calculado sobre los datos de entrenamiento. El algoritmo Mean puede ser utilizado como un punto de referencia sólido para evaluar la eficacia de otro algoritmo frente a un electrodoméstico con un uso de energía disperso.

2.4.2 Edge Detection

El algoritmo de Hart (HART, 1992), es uno de los métodos básicos más utilizados en NILM. Este algoritmo utiliza la detección de bordes en una señal 2D. El término "borde" se refiere a la diferencia de potencia entre dos marcas de tiempo adyacentes, y estos bordes corresponden a estados transitorios y estados

estables de los dispositivos. Aunque el algoritmo es no supervisado, utilizamos datos de entrenamiento para establecer una relación entre los bordes y los dispositivos de manera que se maximice la precisión.

2.4.3 Combinatorial Optimization

La suposición principal en el algoritmo CO es que cada dispositivo puede encontrarse en uno de los estados disponibles (1 de K , donde K es un número pequeño de estados definidos en un dispositivo), y cada estado tiene un consumo de energía asociado.

El objetivo del algoritmo es asignar estos estados a los dispositivos de manera que se minimice la discrepancia entre la lectura agregada del consumo eléctrico de la vivienda y la suma del consumo de energía de los diferentes electrodomésticos. La complejidad temporal de CO aumenta exponencialmente a medida que se incrementa el número de dispositivos y, por lo tanto, no se adapta bien a situaciones en las que hay muchos electrodomésticos.

2.4.4 Discriminative Sparse Coding

Sparse coding estima el uso de cada dispositivo representando el uso agregado como un producto de bases sobredimensionadas y activaciones. Las bases se calculan aprendiendo el consumo general de cada dispositivo de forma individual y luego concatenándolas. Luego se usa un método discriminatorio para obtener las activaciones óptimas.

2.4.5 Exact FHMM, Aproximate FHMMA

Para cada dispositivo se aprende una cadena de markov oculta, lo cual nos devuelve los estados y las probabilidades de transición. Cada estado es asociado a un valor discreto de potencia de operación del dispositivo. Se combinan los modelos de distintos dispositivos para crear una Super Cadena de Markov Oculta. Una extensión es la Aproximate FHMMA, donde se asigna un valor continuo de potencia a los estados.

2.4.6 Denoising autoencoder

Toma una ventana de la señal principal agregada del sistema como entrada de la red y devuelve como salida las lecturas individuales de cada dispositivo por cada

paso de tiempo de la ventana. Esto fue presentado en otra extensión del framework llamada Neural NILM (Kelly & Knottenbelt, 2015).

2.4.7 RNN

Modelo presentado en Neural NILM (Kelly & Knottenbelt, 2015), que usa una LSTM bidireccional para procesar la ventana de tiempo y como salida se tiene un simple valor de consumo de un dispositivo individual.

2.4.8 Seq2Point & Seq2Seq

Seq2Point usa convoluciones de 1 dimensión para procesar la ventana de entrada y devuelve la salida hacia el punto medio de la ventana ingresada, por otro lado, Seq2Seq procesa la lectura sobre toda la ventana ingresada. (Zhang & Zhong, 2018).

2.5 Pronóstico de consumo energético

El pronóstico del consumo energético basado en los datos de un medidor IoT no solo ayuda a gestionar mejor los recursos energéticos, sino que también puede llevar a ahorros significativos en costos y contribuir a un uso más sostenible de la energía.

Debido a las diferencias en las actividades y consumos energéticos de cada hogar, las técnicas tradicionales, como los modelos autorregresivos de media móvil, los métodos de aprendizaje automático y las redes neuronales no profundas, tienen dificultades para proporcionar predicciones precisas del consumo eléctrico de un hogar individual. Estudios recientes han demostrado que las redes neuronales de memoria a corto y largo plazo (LSTM) superan a la mayoría de estos métodos tradicionales en la predicción del consumo de energía. Sin embargo, existen dos desafíos de investigación aún sin resolver en la literatura. En primer lugar, la precisión de las predicciones todavía no cumple con los estándares prácticos para aplicaciones industriales en el mundo real. En segundo lugar, la mayoría de los estudios existentes se enfocan solo en predecir un paso adelante en el tiempo, lo que resulta en pronósticos con un horizonte de tiempo demasiado corto para ser útiles en situaciones prácticas (Yan & Wang, 2018).

Existen algunos trabajos donde proponen modelos novedosos, (Kumar & Gupta, 2023) usa Ensemble Learning para generar las predicciones obteniendo métricas superiores a métodos convencionales. En la siguiente figura se muestra el modelo propuesto.

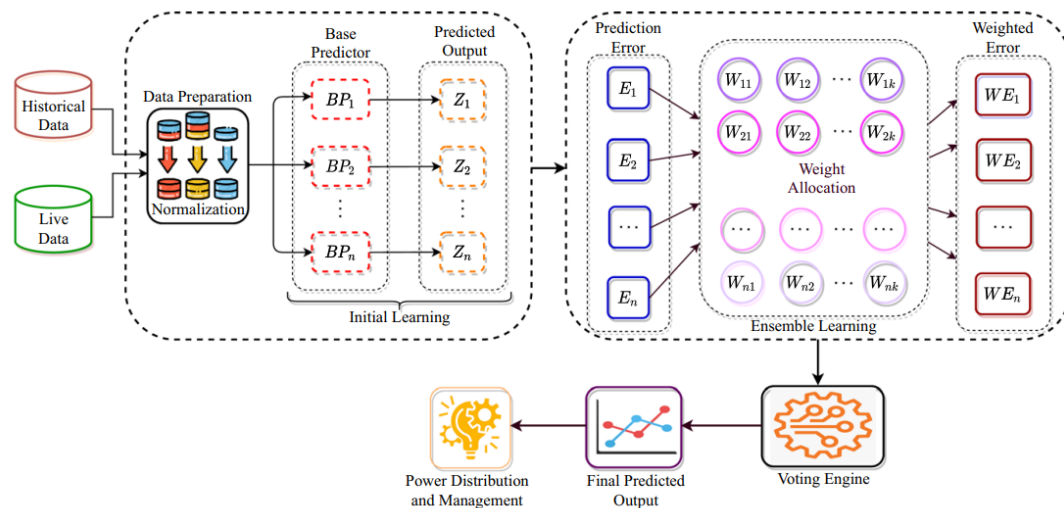


Ilustración 9. Modelo de EPC-PM para pronóstico de consumo energético (Kumar & Gupta, 2023)

Dicho modelo consta de tres partes: entrenamiento inicial, ensemble learning y un motor de votación:

- I. Entrenamiento inicial: Datos históricos y en tiempo real entran al sistema de EPC-PM, se preprocesan los datos para eliminar valores faltantes mediante interpolación y se normalizan. Para construir los predictores bases se pueden usar algoritmos de aprendizaje automático convencionales como *bagging regressor*, *random forest* o regresión lineal.
- II. Ensemble learning: Combina la salida de múltiples predictores bases y su rendimiento es considerado para las siguientes iteraciones. El error también es calculado para afinar las predicciones del siguiente intervalo. Un peso individual es asignado a cada predictor base calculado como el producto entre la predicción de cada modelo el error. Este peso se mantiene hasta la siguiente iteración donde será actualizado bajo el mismo proceso.
- III. Motor de votación: Después de obtener los pesos de cada predictor base, el motor de votación selecciona el mejor predictor base que tenga el peso

mínimo y calcula la predicción final del EPC-PM. El rendimiento del modelo depende del performance de cada predictor base.

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN

En este capítulo se detallarán uno a uno los métodos y procesos realizados para cumplir cada uno de los objetivos específicos y resultados esperados detallados en el primer capítulo.

3.1 Diseño de sistema de telemetría

Para la ingesta de datos y uso en operación del sistema propuesto se necesita la configuración de un medidor de energía IoT con capacidad de enviar los datos obtenidos a una frecuencia deseable y por una vía de comunicación adecuada para nuestro sistema. A continuación, se explicará el proceso de configuración del dispositivo elegido, parámetros de operación, almacenamiento y obtención de datos.

3.1.1 Conexiones físicas para medidores de energía IoT

El lugar designado para la implementación del presente proyecto es el Laboratorio de Sistemas Telemáticos, ubicado en la Escuela Superior Politécnica del Litoral. En dicho espacio, se cuenta con un tablero de Breakers trifásicos, los cuales controlan la distribución de energía al espacio. En este tablero se instalará el dispositivo **Shelly 3EM**. A partir de esta sección a este dispositivo lo referenciamos como **medidor agregado**, puesto que las variables que nos permitirá recolectar corresponden a la suma agregada de todas las cargas eléctricas.

El proceso de instalación consiste en la colocación de 3 transformadores de corriente con una capacidad máxima de 120 A, uno por cada fase. A su vez se instalan 3 pinzas o lagartos en los terminales expuestos de cada fase, más uno en el neutro.

Este dispositivo se configura mediante la aplicación móvil propietaria de la marca Shelly y permite establecer los parámetros operáticos, tales como red de

conexión, fecha, hora, valores de calibración, y lo más importante, el punto de acceso de datos (MQTT CLIENT) que se explicará en las siguientes secciones.

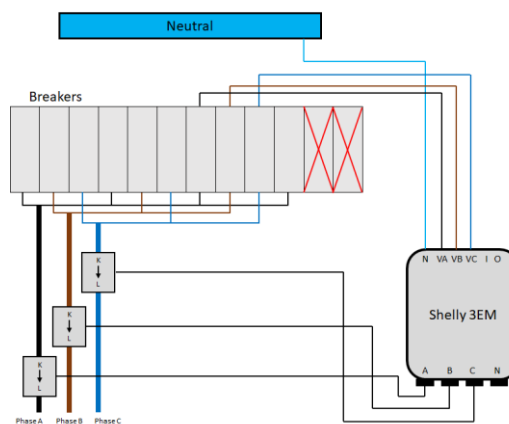


Ilustración 10 Esquema de conexión para el dispositivo Shelly 3EM

Para obtener los datos de cada dispositivo o carga individual de la red, usaremos el medidor **Sonoff S31**, el cual consiste en un plug individual de 1 toma corriente tipo B de tres contactos. A partir de esta sección lo conoceremos como **medidor individual**. Dicho medidor no necesita ninguna instalación eléctrica, se coloca entre el enchufe del dispositivo deseado y el enchufe principal.

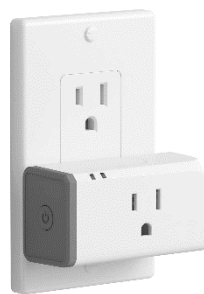


Ilustración 11 Imagen referencial del dispositivo Sonoff S31 (SonOff, 2024)

Para los propósitos de este proyecto se cambia el sistema operativo de fabrica del medidor individual por uno de código libre llamado **TASMOTA**. Este firmware es un proyecto comunitario que permite añadir funcionalidades a dispositivos basados en la arquitectura ESP, específicamente los chipsets ESP8266, ESP32,

ESP32-S y ESP32-C3. La decisión de usar este firmware se justifica en la frecuencia de datos y el método de transmisión de datos.

Con el firmware original del S31, solo se puede hacer una grabación de datos directamente en la aplicación del fabricante, es decir, no se puede transmitir ni enviar datos a una base de datos externa. Con el firmware propuesto se puede configurar arbitrariamente un bróker MQTT para el envío de datos, así como modificar las frecuencias de muestreo y variables deseadas.

Para el cambio de firmware se necesita dismantelar la carcasa original del Sonoff S31 y soldar un adaptador serial-USB a los pines expuestos en la documentación. Mediante un ordenador se establece conexión y se carga el firmware. Al finalizar uno puede configurar los parámetros de operación antes mencionados de una forma similar al del medidor agregado.

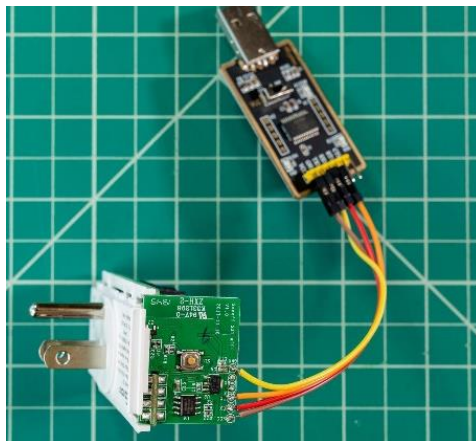


Ilustración 12 Imagen referencial del proceso de comunicación serial-USB para carga de firmware

3.1.2 Configuración lógica para medidores de energía IoT

Para el diseño de la arquitectura IoT que usaremos definimos los siguientes conceptos:

- Sensor/medidor: Dispositivo capaz de enviar información mediante cualquier medio de transmisión encapsulado en un protocolo de comunicación.
- MQTT: Protocolo de transporte de mensajería de publicación y suscripción. Diseñado para el intercambio eficiente de datos en tiempo real. (Soni, 2017)
- Broker MQTT: Sistema Back-end que coordina los mensajes entre los clientes. Tiene como función recibir y filtrar mensajes, identificar a clientes suscritos y retransmitir mensajes.
- Cliente MQTT: Cualquier dispositivo que pueda correr una librería MQTT. Se clasifica en publicadores y suscriptores. Los publicadores envían mensajes a un tema específico. Los suscriptores pueden seleccionar que temas escuchar para recibir mensajes.
- Servidores: Conjunto de equipos informáticos con alta capacidad de procesamiento destinados para el despliegue en operación de software.
- HayIoT: Proyecto de código abierto donde implementa la tecnología Haystack aplicado a dispositivos IoT. Proporciona un marco de trabajo ágil para la creación de dispositivos e ingesta de datos estandarizados.

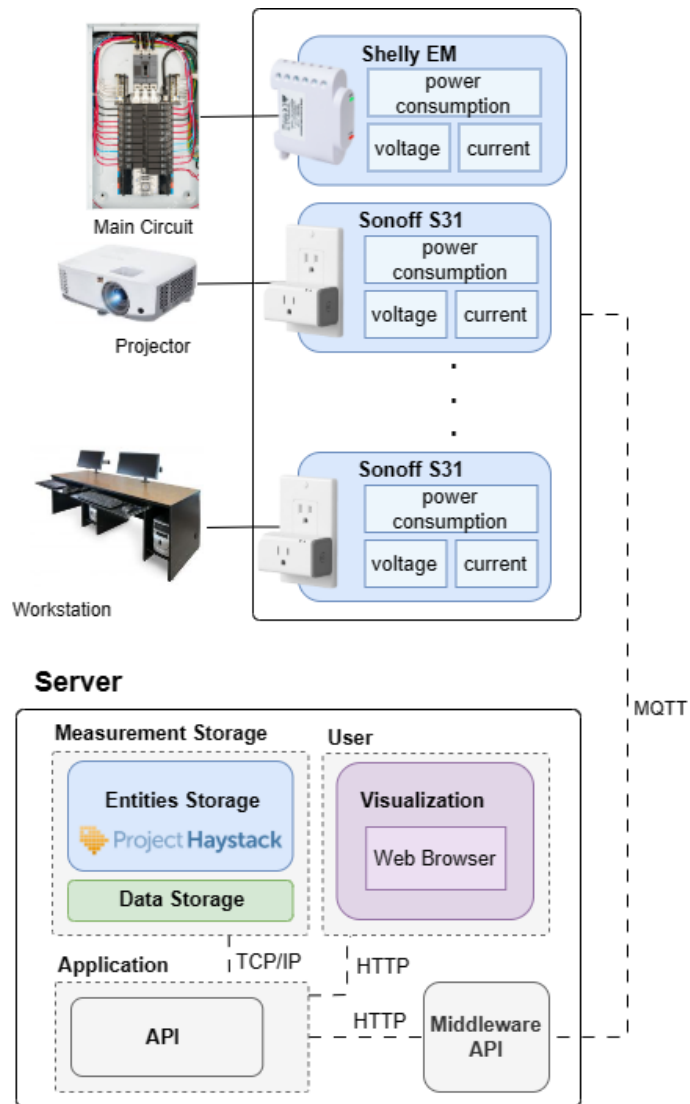


Ilustración 13 Diagrama de arquitectura para telemetría

En la arquitectura propuesta, tanto el medidor agregado como los medidores individuales tienen configurado un Cliente MQTT. Dicho cliente está configurado para transmitir a una frecuencia de 10 segundos.

La ventaja de implementar el proyecto HayIOT es que abstrae el uso de una base de datos. Internamente este proyecto implementa una base de datos no relacional para guardar los campos que se envíen en el payload, lo cual permite escalar añadiendo nuevos dispositivos sin tener que realizar mayor configuración de backend.

El bróker MQTT que se está usando del lado del servidor es Mosquitto, esta elección no viene de algún requerimiento especial, puesto que los brokers comunes que están en el mercado cumplen todos con el mismo propósito.

El proceso de ingesta de datos se resume en el siguiente diagrama:

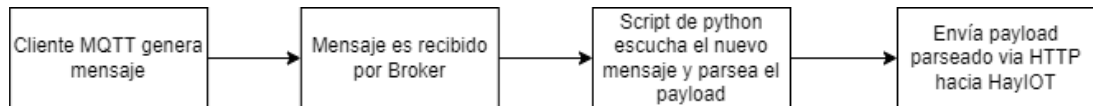


Ilustración 14 Diagrama de conexión MQTT de alto nivel

Existe un script en Python, que también actúa como cliente MQTT en el rol de suscriptor y transforma el mensaje recibido en el formato de los medidores, para luego mediante una petición HTTP ingesta los datos en el sistema HayIoT con el formato mostrado en la siguiente tabla:

Tabla 4. Formato de datos de cada dispositivo IoT en el estudio

| SonOff S31 | Shelly3EM |
|---|--|
| <pre>{ "ApparentPower" : <double>, "ReactivePower" : <double>, "Power" : <double>, "Factor" : <double>, "Voltage" : <double>, "Current" : <double>, }</pre> | <pre>{ "corriente_F" : <double>, "potencia_F" : <double>, "energia_F" : <double>, "factor_potencia_F" : <double>, "voltaje_F" : <double>, }</pre> <p>* Donde F es el numero de la fase a medir</p> |

3.2 Creación de dashboard de monitoreo

Para la elaboración de una plataforma que permita visualizar de forma correcta, útil y amigable con el usuario final, se realizó una encuesta con distintos tipos de

usuarios donde se definieron historias de usuario, lo que permitió en el posterior desarrollo de módulos para visualización de información.

3.2.1 Historias de usuario

Tabla 5. Descripción de Hito 1

| |
|--|
| Hito 1: Accesibilidad y compatibilidad web |
| Como usuario quiero poder acceder a la plataforma de visualización a través de una tecnología web responsive que permita explorar las distintas funcionalidades y me permita observar y obtener insights de valor. |

Tabla 6. Descripción de Hito 2

| |
|--|
| Hito 2: Configuración de periodos |
| Como usuario quiero poder configurar el periodo de facturación que se ha asignado a mi vivienda/oficina que permita llevar con exactitud el registro de consumo. |

Tabla 7. Descripción de Hito 3

| |
|--|
| Hito 3: Configuración de notificaciones |
| Como usuario quiero poder configurar notificaciones para cuando estoy superando los limites o cuotas de uso. |

Tabla 8. Descripción de Hito 4

| |
|---|
| Hito 4: Ver el consumo en kWh |
| Como usuario quiero visualizar el consumo acumulado en kWh que se registra en el periodo actual que permita contrastar con los valores finales de la factura eléctrica. |

Tabla 9. Descripción de Hito 5

| |
|--|
| Hito 5: Ver el consumo en \$ |
| Como usuario quiero poder visualizar el consumo actual en su equivalente en dinero que permita contrastar con los valores finales de la factura eléctrica. |

Tabla 10. Descripción de Hito 6

| |
|--|
| Hito 6: Predicción de consumo |
| Como usuario quiero un método para visualizar cuanto consumiré al final de periodo que permita tomar acciones correctivas en caso de que este fuera de los límites de operación. |

Tabla 11. Descripción de Hito 7

| |
|---|
| Hito 7: Consumo en tiempo real de métricas |
| Como usuario quiero visualizar valores como el consumo instantáneo, voltaje, corriente que permita identificar alguna anomalía en la operación eléctrica. |

Tabla 12. Descripción de Hito 8

| |
|---|
| Hito 8: Desagregación de dispositivos |
| Como usuario quiero poder visualizar los dispositivos individuales que tengo conectado que permita identificar los patrones de con. |

3.2.2 Definición de módulos del dashboard

A continuación, se detalla individualmente cada una de las vistas realizadas para el dashboard.

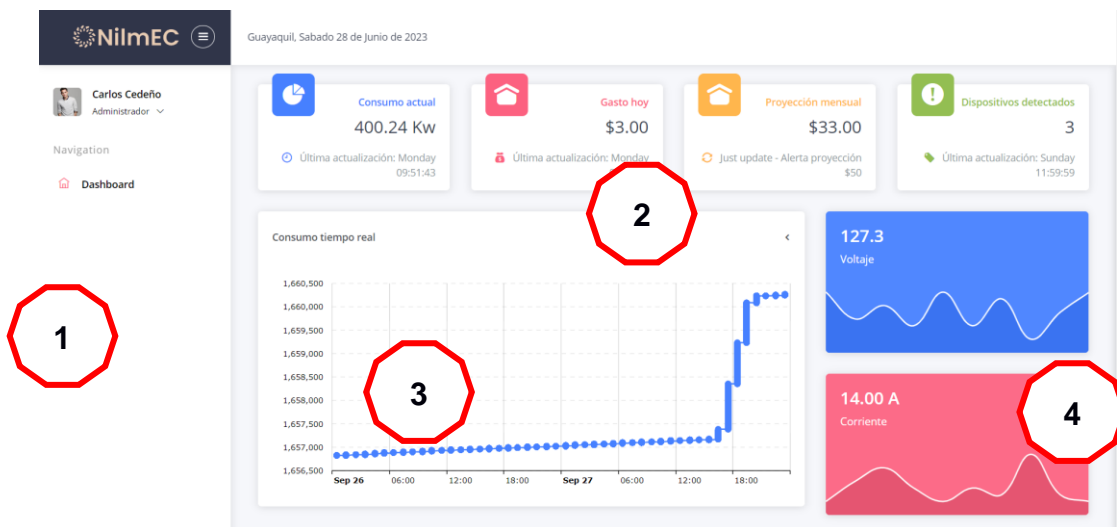


Ilustración 15 Vista general del dashboard

1. Panel de navegación: En este panel se muestran las secciones de la página. Consta de 3 secciones, Dashboard, Configuración y alertas. Cuenta con un menú búrguer que despliega el menú de forma lateral y es responsive para pantallas de formato vertical.
2. Paneles de métricas: Aquí se muestran algunas métricas de importancia como el consumo total en KW, consumo en equivalente monetario, la proyección de gasto mensual al final del periodo y el número de dispositivos desagregados.
3. Panel en tiempo real 1: Muestra el consumo en tiempo real en kWh.
4. Panel en tiempo real 2: Muestra indicadores de la operación de la red en tiempo real.

3.3 Perfil de carga de dispositivos

En esta sección se describen los distintos procesos y técnicas que se usaron para generar un perfil de carga que caracterice los estados de operación de los dispositivos mencionados.

Como se mencionó en el capítulo anterior, un perfil de carga de dispositivo eléctrico se refiere a la descripción o representación gráfica de cómo un dispositivo eléctrico consume energía eléctrica a lo largo del tiempo. Este perfil de carga puede mostrar

cómo varía la potencia o la corriente eléctrica que el dispositivo utiliza en diferentes momentos del día o durante un período específico.

Para nuestra aplicación de NILM, el principal atributo que deseamos conocer sobre las cargas eléctricas en el sistema es la potencia(s) de operación del dispositivo. La potencia de operación puede variar según el dispositivo y la carga que está manejando. Algunos dispositivos tienen una potencia constante, mientras que otros pueden variar su potencia dependiendo de la carga o el modo de funcionamiento.

A continuación, analizaremos distintos métodos para obtener esta potencia de operación para uno de los dispositivos de nuestro estudio, luego en el capítulo de resultados presentaremos el análisis para todos los dispositivos.

El dispositivo seleccionado es un Proyector Marca Epson Modelo PowerLite W39.

3.3.1 Método visual – Histograma.

Los dispositivos eléctricos usualmente indican la potencia nominal que consumirán en su manual de usuario, pero como se explicó en el capítulo 2, esta potencia no es constante puesto que existen estados de operación que pueden o no afectar directamente la potencia que se está consumiendo. En este dispositivo claramente se pueden identificar 3 estados:

1. Apagado: El dispositivo no recibe ni proyecta señal alguna. La lámpara del dispositivo está apagada. Solo recibe el comando de ON.
2. Suspensión / Ahorro de energía: El dispositivo presenta una pantalla azul en dos situaciones, no existe una fuente o señal de video entrante, o ha pasado 1 hora sin cambios en la señal de video. Este tiempo es configurable y su existencia depende del fabricante.
3. Proyección: Proyecta constantemente una señal de video.



Ilustración 16 Modo suspensión - Ahorro de energía

Para comprobar lo anteriormente dicho, usaremos una técnica de histograma para mediante el conteo de frecuencias identificar que valores son los más comunes. Se tomaron los valores donde $P_{activa} > 0$, dado que el estado más común del dispositivo por su comportamiento de uso (derivado del comportamiento humano) es el estado apagado.

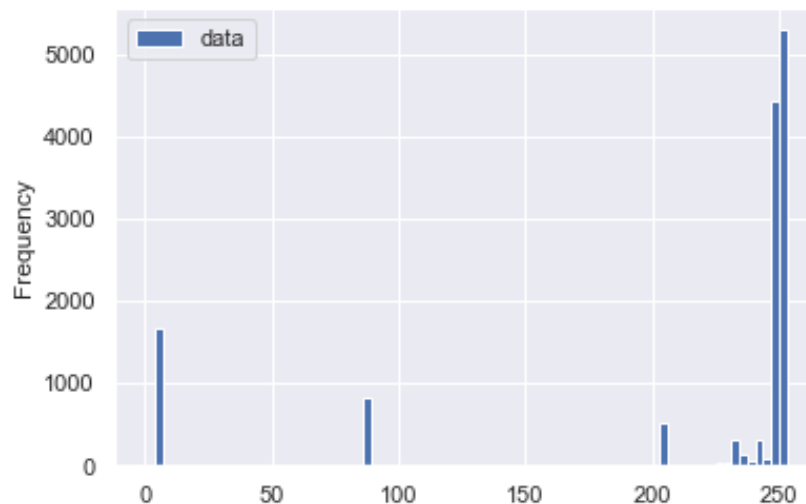


Ilustración 17 Histograma de la Potencia activa. Bins=80

Del grafico podemos observar claramente 4 posibles clústeres, uno más de lo que nuestro diagnóstico manual identificó. Por lo cual según la literatura usaremos otros

métodos con mayor poder explicativo que nos permitan realizar un correcto análisis de los datos.

3.3.2 K-means clustering

K-means es un algoritmo de clasificación no supervisada (clusterización), que busca dividir un conjunto de datos en k grupos o clústeres, minimizando la suma de las distancias al cuadrado entre los elementos y los centroides de sus respectivos grupos. (Hastie & Friedman, 2001).

Usando la librería Scikit-Learn para Python, encontraremos primero el número de clústeres que explican mejor los grupos usando la métrica de inercia. La inercia en el contexto del algoritmo K-means se refiere a una medida que se utiliza para evaluar la calidad de los grupos (clústeres) creados por el algoritmo. La inercia se calcula como la suma de las distancias al cuadrado entre cada punto de datos en un grupo y el centroide de ese grupo. En otras palabras, la inercia mide cuán cerca están los puntos de datos dentro de un grupo de su centroide.

El objetivo principal del algoritmo K-means es minimizar esta inercia, ya que un valor de inercia menor indica que los puntos de datos dentro de un grupo están más cerca entre sí y, por lo tanto, el agrupamiento es más coherente.

Cuando aplicamos este concepto a nuestro conjunto de datos, mediante el método del codo, visualmente podemos determinar que $k = 3$ es un valor que podría considerarse óptimo para nuestro análisis, lo que será discutido en el siguiente capítulo.

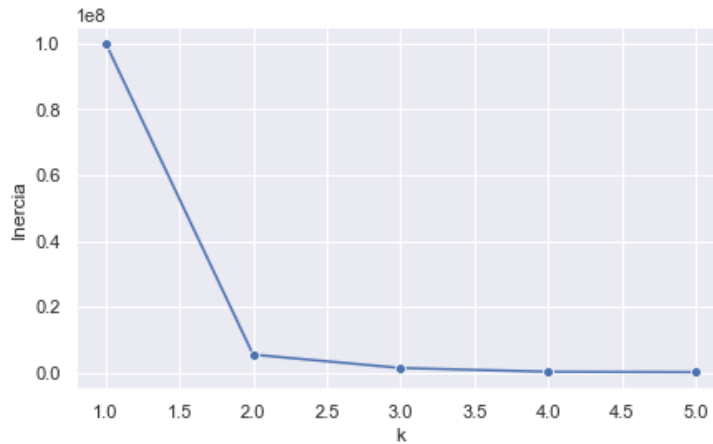


Ilustración 18. Método de inercia para encontrar K óptima

En la siguiente tabla se presentan los resultados para Kmeans.

Tabla 13. Resultados de clustering para k=3 y K=4

| $K = 3$ | $K = 4$ |
|---------|---------|
| 5.93 | 5.93 |
| 89.68 | 88.81 |
| 246.16 | 204.10 |
| - | 248.60 |

3.3.1 Gaussian Mixture Models para clustering

Los modelos GMM (Gaussian Mixture Models) son un enfoque de modelado probabilístico que se utiliza para la clusterización de datos. Estos modelos consideran que los datos provienen de una mezcla de varias distribuciones gaussianas, lo que les permite capturar la complejidad de los clústeres con diferentes formas y tamaños. Se ajustan a los datos estimando parámetros como la media, la varianza y los pesos relativos de cada componente gaussiano.

Estos modelos proporcionan una poderosa herramienta para la agrupación de datos, permitiendo la modelización de estructuras de clúster de naturaleza más compleja que las simples distribuciones unimodales.

Usaremos la librería Scikit-Learn para Python, donde definimos el número de componentes a encontrar y el máximo número de iteraciones del modelo. Como resultado obtenemos el centro de cada componente, la covarianza y el peso de cada componente. A continuación, se muestran estos resultados para $K = 3, 4$

Tabla 14. Resultados estadísticos para $k=3$ y $K=4$

| K | Medias | Covarianzas | Pesos |
|---|--------|-------------|--------|
| 3 | 5.76 | 0.2 | 0.12 |
| | 87.71 | 294.8 | 0.06 |
| | 246.19 | 112.1 | 0.81 |
| 4 | 5.77 | 0.21 | 0.12 |
| | 88.03 | 0.11 | 0.059 |
| | 217.58 | 1556.1 | 0.1243 |
| | 249.58 | 0.79 | 0.69 |

3.3.2 Estimaciones con Kernel Density

En la biblioteca scikit-learn (sklearn) de Python, la función KernelDensity se utiliza para estimar y modelar la densidad de probabilidad de un conjunto de datos utilizando un estimador de kernel. La estimación de densidad de kernel es una técnica no paramétrica que estima la función de densidad de probabilidad (PDF) subyacente de un conjunto de datos.

KDE en sí misma no se utiliza típicamente como una técnica de clustering directa, ya que su objetivo principal es modelar y estimar la distribución de probabilidad subyacente de un conjunto de datos, en lugar de agrupar puntos de datos en clústeres específicos.

Sin embargo, KDE puede ser útil en tareas de clustering como una herramienta de visualización, inicialización de centroides o detección de anomalías.

A continuación, se muestra un ajuste realizado para analizar al menos 4 clústeres derivados de los datos originales de entrada.

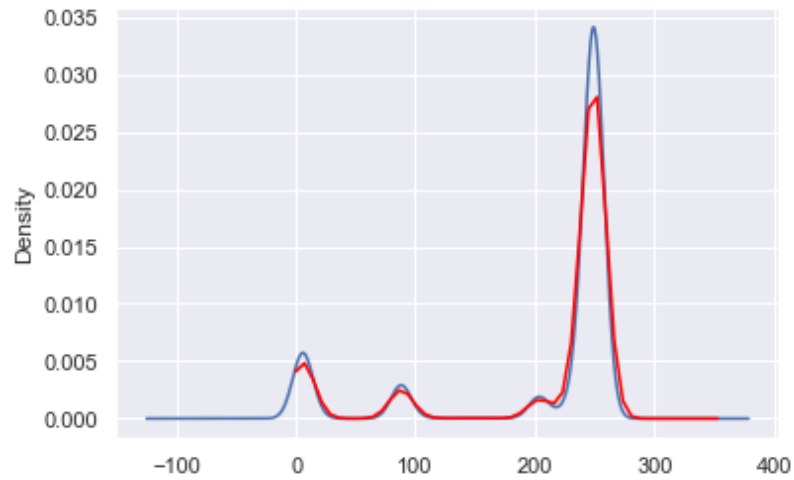


Ilustración 19 Ajuste de curva mediante el método KDE

Se puede observar cómo se presentan cerca de 4 picos que podríamos considerar clústeres en la data original mostrada. Sobre esta nueva curva podemos encontrar centroides, medias, mínimos, máximos y pesos.

Tabla 15. Resultados de método KDE para obtención de clústeres

| Cluster | Muestras | Media | Mediana | Std | Max | Min | Peso |
|---------|----------|--------|---------|-------|-----|-----|--------|
| 1 | 1709 | 5.93 | 6 | 1.49 | 24 | 1 | 0.12 |
| 2 | 880 | 88.43 | 99 | 3.07 | 120 | 61 | 0.063 |
| 3 | 18 | 146.61 | 148.5 | 9.28 | 162 | 131 | 0.0012 |
| 4 | 11271 | 246.15 | 249 | 10.71 | 253 | 167 | 0.812 |

Se observa que los centroides 1, 2 y 4 corresponden a los mismos resultados obtenidos por los otros métodos. El clúster 3 se lo puede descartar por tener muy poca representatividad de la muestra total, menos del 1%.

3.4 Desarrollo de modelo de pronóstico de consumo y facturación de energía eléctrica

En este trabajo se desea obtener los siguientes valores:

- Consumo actual en kW: La potencia activa instantánea que se está consumiendo. Esto nos da una idea de cuanta es nuestra carga eléctrica en un determinado instante de tiempo.
- Consumo acumulado en USD: Siguiendo la metodología explicada en los párrafos anteriores, se obtiene el valor en USD consumido hasta el instante de la consulta.
- Consumos históricos en USD: Bajo la misma metodología, se quiere mostrar un gráfico que ilustre cómo ha evolucionado el consumo en USD los últimos tres meses.
- Pronóstico de consumo en USD: Mostrar un pronóstico de la facturación en USD a finalizar el mes. Para esto, se muestra al modelo n días transcurridos del mes.

La data usada este objetivo proviene de la señal agregada del medidor de consumo **Shelly 3EM**. Esta data contiene la información mostrada en la Tabla 2, donde la variable de interés es *Power* o Potencia Activa.

Como se mencionó en la sección 3.1.1, el panel de distribución de energía es trifásico, quiere decir que cada fase de forma independiente contribuye al consumo energético total del laboratorio. En la Ilustración 20 se pueden observar las conexiones físicas del medidor que adquiere esta información.



Ilustración 20. Conexiones de fase para el medidor Shelly 3EM

La facturación mensual se obtiene multiplicando la energía total consumida en el mes por la tarifa de energía eléctrica, generalmente medida en unidades monetarias por kilovatio-hora (kWh). Dado que nuestras mediciones son de potencia instantánea debemos realizar un procedimiento para calcular su equivalente en kWh.

La energía es la integral de la potencia respecto al tiempo. Para obtener la energía consumida en un período específico, se integra la potencia instantánea sobre ese período. Matemáticamente, la energía E consumida durante un intervalo de tiempo $[t_1, t_2]$ se calcula como:

$$E = \int_{t_1}^{t_2} P(t)dt \quad (3)$$

En la práctica, esta integral se aproxima mediante una suma discreta si los datos de potencia se registran en intervalos de tiempos discretos, donde Δt es el intervalo de tiempo entre dos mediciones sucesivas y N es el número total de mediciones en el periodo considerado:

$$E \approx \sum_{i=1}^N P(t_i)\Delta t \quad (4)$$

Para obtener la energía consumida en un mes, se suman las energías calculadas para todos los intervalos de tiempo que cubren ese mes. Si los datos de potencia

instantánea se registran cada hora, por ejemplo, la energía total mensual $E_{mensual}$ se puede calcular como:

$$E_{mensual} = \sum_{j=1}^M \left(\sum_{i=1}^{N_j} P(t_{i,j}) \Delta t \right) \quad (5)$$

Donde M es el número de días en el mes (o periodo deseado) y N_j es el número de mediciones en el día j . Por último, si la tarifa de energía es T_f en USD/kWh, la facturación mensual se calcula como:

$$Facturacion = E_{mensual} \times T_f \quad (6)$$

En Ecuador, la tarifa eléctrica se define en el Pliego Tarifario del Servicio Público de Energía Eléctrica, para el año 2024 se rige bajo la resolución Nro. ARCERNNR-034/2023 y se definen esta tarifa diferenciada por actividad o uso, tipo de consumidor y franjas horarias como se muestra un fragmento en la siguiente ilustración.

PERIODO: ENERO - DICIEMBRE

EMPRESAS ELÉCTRICAS:

CNEL EL ORO - CNEL ESMERALDAS - CNEL GUAYAS LOS RÍOS - CNEL LOS RÍOS - CNEL MANABÍ - CNEL MILAGRO - CNEL SANTA ELENA - CNEL SANTO DOMINGO - CNEL SUCUMBÍOS - GALÁPAGOS

CARGOS TARIFARIOS ÚNICOS

| RANGO DE CONSUMO | DEMANDA (USD/kW-mes) | ENERGÍA (USD/kWh) | COMERCIALIZACIÓN (USD/Consumidor) | |
|---|----------------------|-------------------|-----------------------------------|--|
| BAJO VOLTAJE CON DEMANDA HORARIA | | | | |
| COMERCIALES | | | | |
| 08:00 hasta 22:00 horas | 4,790 | 0,090 | 1,414 | |
| 22:00 hasta 08:00 horas | | 0,072 | | |
| INDUSTRIALES | | | | |
| 08:00 hasta 22:00 horas | 4,790 | 0,065 | | |
| 22:00 hasta 08:00 horas | | 0,069 | | |
| E. OFICIALES, ESC. DEPORTIVOS SERVICIO COMUNITARIO Y ABONADOS ESPECIALES | | | | |
| 08:00 hasta 22:00 horas | 4,790 | 0,080 | | |
| 22:00 hasta 08:00 horas | | 0,066 | | |
| BOMBEO AGUA | | | | |
| 08:00 hasta 22:00 horas | 4,790 | 0,070 | | |
| 22:00 hasta 08:00 horas | | 0,056 | | |

Ilustración 21. Fragmento tomado del Pliego Tarifario del Servicio Público de Energía Eléctrica 2024

Para simplificar , a partir de este punto consideraremos esta tarifa como 0.06 USD/kWh. Una vez que ya tenemos este proceso programado en un script, podemos enviarle un conjunto de datos (fecha/hora con valor de potencia activa) y obtendremos como resultado el consumo de ese periodo en USD.

Estos datos de consumo se consideran como una serie temporal. Son una secuencia de datos ordenados cronológicamente a intervalos iguales o desiguales. El proceso de pronóstico consiste en predecir el valor futuro de una serie temporal, ya sea modelando la serie únicamente en función de su comportamiento pasado (autorregresivo) o utilizando otras variables externas. En el presente estudio se desea realizar el pronóstico de consumo energético mensual a partir de los datos históricos, esto con el fin de proporcionar una guía al usuario de cómo se desarrolla su consumo y los potenciales efectos al final del periodo facturable.

Para este trabajo se decidió usar un modelo recientemente desarrollado por el equipo de *Amazon Science* llamado *Chronos*.

Chronos es una familia de modelos de predicción de series temporales pre-entrenados, basados en arquitecturas de modelos de lenguaje. La serie temporal se transforma en una secuencia de tokens mediante escalado y cuantificación, y un modelo de lenguaje se entrena con estos tokens usando la pérdida de entropía cruzada (Ansari, 2024). Una vez entrenado, se obtienen pronósticos probabilísticos mediante el muestreo de múltiples trayectorias futuras dadas en el contexto histórico.

Los modelos *Chronos* han sido entrenados con una gran cantidad de datos de series temporales disponibles públicamente, así como en datos sintéticos generados mediante procesos gaussianos. Este modelo se define como “zero-shot”, se refiere a la capacidad de los modelos para generar pronósticos a partir de conjuntos de datos no vistos. Los desarrolladores reportaron un rendimiento superior sobre los modelos de predicción estadísticos clásicos y los modelos de aprendizaje profundo especializados, después de una evaluación exhaustiva en más de 40 conjuntos de datos.

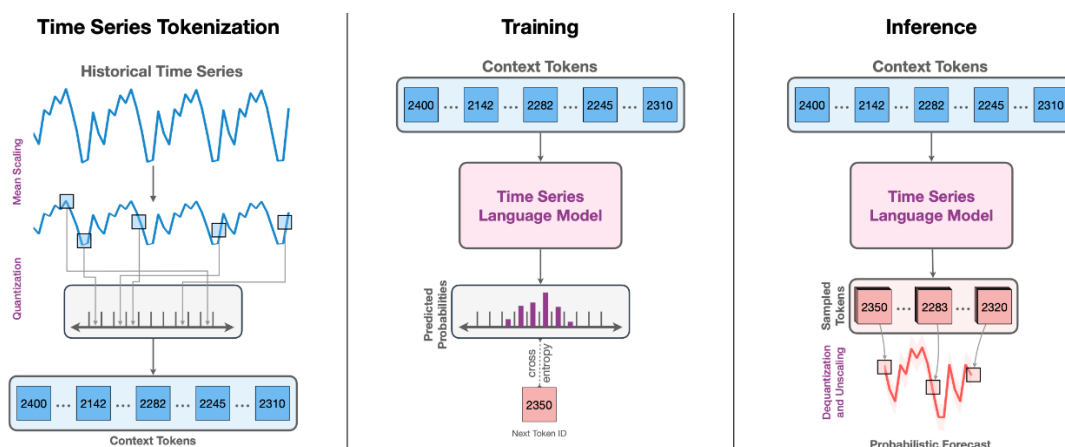


Ilustración 22. Representación a alto nivel de Chronos (Ansari, 2024)

Como se muestra en la Ilustración 22, la serie temporal de entrada se escala y cuantifica para obtener una secuencia de tokens. Luego, los tokens se ingresan en

un modelo de lenguaje, que puede ser un modelo *encoder-decoder* o solo *decoder*. Durante la inferencia, se muestrea auto-regresivamente tokens del modelo y se mapean de nuevo a valores numéricos. Adicional, se muestrean múltiples trayectorias para obtener una distribución predictiva.

En el capítulo 4 evaluaremos los resultados de usar esta novedosa técnica de pronóstico sobre los datos del sistema de telemetría.

3.5 Desarrollo de modelo de desagregación no intrusivo de cargas

Obtener la desagregación de la carga eléctrica es crucial para comprender el consumo de energía dentro de un sistema. Esto nos brinda una interpretación precisa de los patrones de consumo al segmentar el consumo total en componentes más pequeños, como electrodomésticos o dispositivos individuales.

Estos datos son indispensables para la gestión de la energía, incluidas las iniciativas de ahorro de energía, la optimización de la eficiencia, la planificación de infraestructuras y la implementación de políticas de gestión de la demanda. En última instancia, la desagregación ayuda a los consumidores a tomar decisiones informadas sobre el consumo de energía, mejorando la eficiencia y la sostenibilidad de los sistemas eléctricos.

Para este trabajo partimos del framework NILMTK, que significa Non-Intrusive Load Monitoring Toolkit (Kit de Herramientas para el Monitoreo de Carga No Intrusivo), el cual es una biblioteca de código abierto diseñada para facilitar la investigación en el campo del monitoreo de carga no intrusivo. Se usa este proyecto ya que es un proyecto de código abierto que impulsa la estandarización en los métodos y algoritmos disponibles para la investigación en el campo NILM. A continuación, se describen los beneficios que se destacan por usar este framework:

1. **Investigación y Desarrollo:** Proporciona un entorno robusto y flexible para el desarrollo y la evaluación de algoritmos de NILM. Investigadores y desarrolladores pueden utilizar NILMTK para comparar el rendimiento de diferentes algoritmos de desagregación de energía.

2. **Estandarización:** Ofrece una plataforma estandarizada para evaluar algoritmos de NILM, lo que facilita la replicación y comparación de resultados en la comunidad de investigación.
3. **Análisis de Datos:** Incluye herramientas para la manipulación y el análisis de datos de consumo de energía, permitiendo obtener estadísticas e indicadores importantes de forma rápida.
4. **Visualización:** Proporciona herramientas de visualización que ayudan a comprender mejor los patrones de consumo de energía y los resultados de los algoritmos de desagregación.
5. **Extensibilidad:** Es una plataforma extensible, lo que permite a los usuarios añadir sus propios algoritmos, conjuntos de datos y métricas de evaluación.

Se destaca que existen varios datasets disponibles para NILM que ya vienen implementados para su utilización directamente en NILMTK, los cuales fueron mencionados en el capítulo 2.3.

El proceso para aplicar NILM en este proyecto consta de 3 fases: recolección, procesamiento, entrenamiento, análisis e inferencia. Estas fases se realizan de manera secuencial en la primera iteración, con opción a reentrenar el modelo de acuerdo con las necesidades emergentes del lugar de aplicación, por ejemplo, añadir un nuevo dispositivo en la lista de dispositivos soportados o reentrenar con data actualizada.

A continuación, se detallan cada una de las fases:

- I. **Recolección:** a partir de lo detallado en la metodología de la sección 3.1, se tiene en la base de datos los datos de los equipos enviados mediante el modelo de telemetría descrito. En esta fase se descarga de forma automatizada la data

entre dos fechas específicas y se guardan en formato CSV en una carpeta para su posterior procesamiento. Esta fase suele demorar entre 30 minutos a 45 minutos para un solo archivo (un dispositivo) con 4 meses de datos, cerca de 1.1 millones de registros.

II. **Procesamiento:** Con los archivos descargados localmente, se deben convertir en formato HDF5, para lo cual primero se deben localizar los campos de interés y eliminar los campos no importantes para nuestro estudio. Siguiendo la metodología de NILMTK, se deben escribir los archivos de metainformación que describan los medidores, los sitios y el dataset, para lo cual existe el proyecto NILM-METADATA (Kelly J. a., 2014) .Estos archivos se escriben en formato YAML y contienen la siguiente información.

- *Building:* Este archivo describe los dispositivos, además su relación con los medidores, es decir que medidor registra uno o más dispositivos, en qué fase, si hay medidores agregados, etc.
- *Meter devices:* En este archivo se definen las características físicas de los medidores, así como los tiempos de muestreo, limitantes de software, modelos, capacidades de comunicación, etc.
- *Dataset:* Este archivo contiene la descripción del dataset, institución que lo realiza, zona horaria, información de geolocalización, y demás información relevante sobre los autores del dataset.

Esta fase termina con la creación del archivo HDF5 usando un convertidor. Para este estudio se planea usar un convertidor basado en el dataset publico AMPDS.

III. **Entrenamiento:** En esta fase, se definen las fechas para seleccionar data valida de nuestro dataset, se definen los parámetros sobre los cuales se va a aplicar el entrenamiento, los modelos sobre los cuales se van a entrenar y su configuración de hiperparámetros. Esto se facilita mediante el uso de NILMTK-API, el cual es una forma de describir toda la información del “experimento” mediante un objeto JSON. Esto facilita de la replicabilidad de experimentos y

proporciona un mecanismo fácil para editar los parámetros antes mencionado. En el siguiente capítulo se muestra un experimento a profundidad.

- IV. **Análisis e Inferencia:** Una vez entrenados los modelos, se obtienen las métricas de desempeño sobre un conjunto de datos definidos en el objeto de NILMTK-API, estas métricas nos permiten elegir cuál de los modelos realizó mejor las tareas de desagregación. Elegido el modelo, se realiza la exportación de los pesos en formato KERAS para su posterior uso en la aplicación planteada en la sección 3.2.

Para este trabajo, las fases mencionadas I, II y III no se realizan periódicamente, mientras que la fase IV se debe implementar en tiempo real en la aplicación web propuesta para la desagregación del consumo energético. En la Ilustración 23 se puede observar el flujo de todo el proceso entre las fases ya mencionadas.

En el siguiente capítulo se revisarán los datos creados como metadata del dataset, así como un benchmark de algunos modelos sobre los dispositivos del dataset propuesto y el pipeline final del aplicativo web.

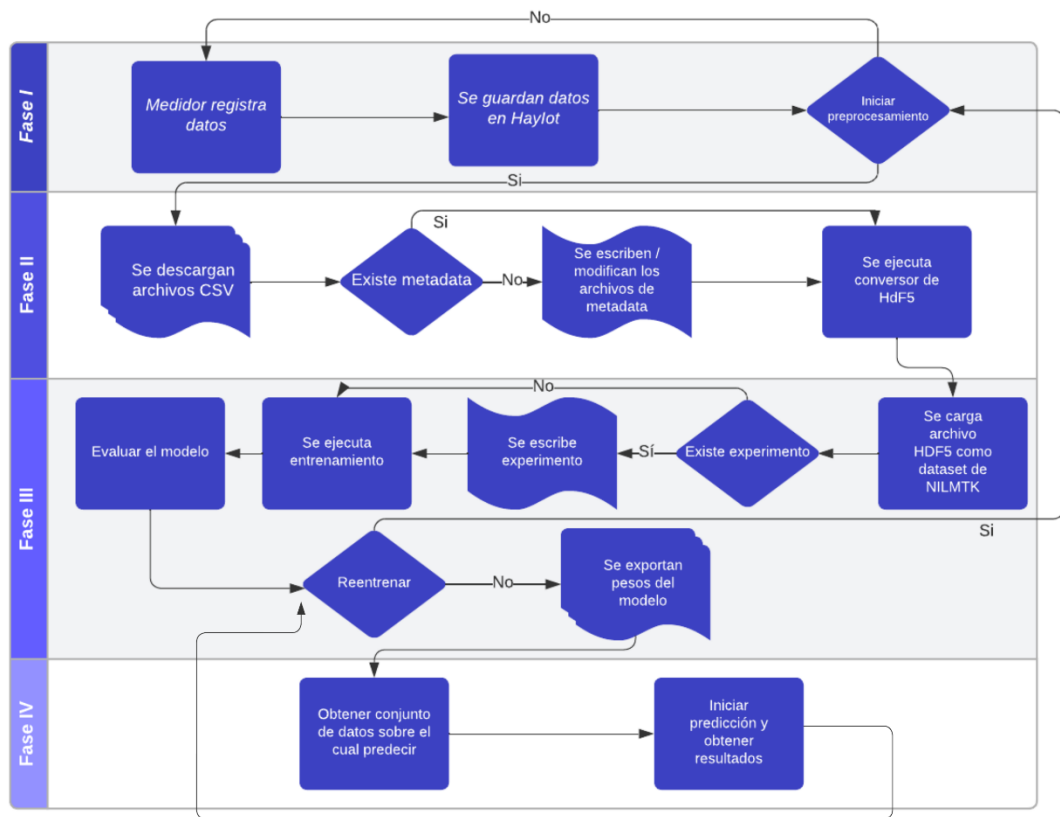


Ilustración 23 Diagrama de procesos para la generación del modelo de desagregación de cargas eléctricas

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

En este capítulo se detallarán los resultados obtenidos para cada una de las propuestas detalladas en el capítulo 1. Estos resultados incluyen un análisis enfocado a las probables causas y efectos que tiene el comportamiento de los dispositivos en cada una de las interrogantes planteadas.

4.1 Dataset para desagregación de cargas

Luego adquirir los equipos de desagregación de cargas individuales y aplicar el procedimiento explicado en la sección 3.1.1, se pudieron ubicar los sensores de acuerdo con la siguiente tabla.

Tabla 16. Lista de dispositivos incluidos en el estudio

| Ubicación | Dispositivo/ID | Carga eléctrica |
|-----------|----------------|-----------------------|
| LST | s31_01 | PC0 |
| LST | s31_02 | Monitor |
| LST | s31_03 | Proyector |
| LST | s31_05 | Router |
| LST | s31_06 | Switch |
| LST | s31_07 | PC1 |
| LST | s31_08 | PC2 |
| LST | s31_09 | PC3 |
| LST | s31_10 | PC4 |
| LST | S31_11 | PC5 |
| LST | Shelly_A | AC1 |
| LST | Shelly_B | AC2 |
| LST | Shelly | Carga total trifásica |

Estos dispositivos recolectaron data desde el 16/06/23 hasta el 20/07/2024, data que ha sido usada para realizar este análisis. Asimismo, la data energética agregada del sitio fue recolectada desde el periodo 11/07/23 hasta el cierre de este estudio.

Dado que para armar construir el dataset necesitamos ambas, las mediciones agregadas e individuales, el periodo real de nuestro dataset corresponde a *400 días* de mediciones cada 10s. Para exportarlo en formato .CSV esto ocupa alrededor de *400MB* por dispositivo y *1500MB* por data agregada, dando un total de *6.3 GB* de información. Esto sin embargo puede reducirse considerando los siguientes factores:

- A) Usar un formato de exportación menos pesado. Existen formatos como HDF5 (Hierarchical Data Format 5) el cual es un formato de archivo diseñado para almacenar y organizar grandes volúmenes de datos de manera eficiente y flexible.
- B) Reducir la información redundante o innecesaria. Por ejemplo, el número de medición o el identificador del dispositivo. Esta data puede compartirse de forma externa al dataset sin que afecte la verdadera utilidad de este.
- C) Reducir la cantidad de variables. Dado a que este análisis va enfocado a procesos eléctricos toda variable que sea incluida debe aportar valor, sin embargo, como este es un dataset pensado en la aplicación NILM, existen variables como el factor de potencia que no aportan utilidad.

Nombrados los factores anteriores, se justifica hacer el desarrollo del dataset con el formato de NILMTK, usando los lineamientos de NILMTK-METADATA.

4.1.1 Creación del dataset en formato HDF5 para NILMTK

Como se mencionó en el capítulo 3.5, se deben crear los archivos de metadata que incluye la definición del dataset, la descripción de los medidores y dispositivos y la descripción del sitio de estudio. A continuación, se muestran dichos archivos.

dataset.yml

Información general del dataset.

```
name: ULLED
long_name: Computer Lab Level Energy Dataset
creators:
- Cedeno, Carlos
publication_date: 2024
institution: Escuela Superior Politecnica del Litoral (ESPOL)
contact: caencede@espol.edu.ec
description: Several weeks of power data from June 2023 to July
2024.
number_of_buildings: 1
timezone: America/Guayaquil
geo_location:
  locality: Guayas, Guayaquil # village, town, city or state
  country: EC # standard two-letter country code defined by ISO
3166-1 alpha-2
  latitude: -2.1733 # MIT's coordinates
  longitude: -79.9228
schema: https://github.com/nilmtnk/nilm_metadata/tree/v0.2
```

meter_devices.yml

Información específica sobre los medidores como marca, modelo, limitantes físicas.

```
sonoff:
  model: s31
  manufacturer: Sonoff
  manufacturer_url: https://sonoff.tech/
  description: >
    Sonoff Basic es un conmutador inteligente WiFi asequible que
proporciona a
    los usuarios un control inteligente en el hogar. Es un
interruptor de control remoto que
    se puede conectar a una amplia gama de electrodomésticos.
```

```
sample_period: 10 # the interval between samples. In seconds.
max_sample_period: 60 # Max allowable interval between
samples. Seconds.
measurements:
- physical_quantity: power # power, voltage, energy, current?
  type: active # active (real power), reactive or apparent?
  upper_limit: 5000000
  lower_limit: 0
- physical_quantity: voltage # power, voltage, energy,
current?
  type:
  upper_limit: 5000000
  lower_limit: 0
- physical_quantity: current
  type:
  upper_limit: 5000000
  lower_limit: 0
wireless: true

shelly:
model: 3EM
manufacturer: Shelly
manufacturer_url: https://www.shelly.com/
description: >
  Smart 3-phase energy meter with contactor control
sample_period: 10 # the interval between samples. In seconds.
max_sample_period: 60 # Max allowable interval between
samples. Seconds.
measurements:
- physical_quantity: power # power, voltage, energy, current?
  type: active # active (real power), reactive or apparent?
  upper_limit: 5000000
  lower_limit: 0
```

```
- physical_quantity: voltage      # power, voltage, energy,
current?
  type:
  upper_limit: 5000000
  lower_limit: 0
- physical_quantity: current      # power, voltage, energy,
current?
  type:
  upper_limit: 5000000
  lower_limit: 0
wireless: true
```

building1.yml

Información específica sobre el sitio de medición, así como la disposición de las fases, puntos de carga, equipos y dispositivos.

```
instance: 1 # this is the first building in the dataset
elec_meters: # a dictionary where each key is a meter instance
  1:
    site_meter: true # phase_a
    device_model: shelly
  2:
    site_meter: true # phase_b
    device_model: shelly
  3:
    site_meter: true # phase_c
    device_model: shelly
  4:
    submeter_of: 3 # s31_01
    device_model: sonoff
  5:
    submeter_of: 3 # s31_02
    device_model: sonoff
  6:
    submeter_of: 3 # s31_03
```

device_model: sonoff

7:

submeter_of: 3 # s31_05

device_model: sonoff

8:

submeter_of: 3 # s31_06

device_model: sonoff

9:

submeter_of: 2 # s31_07

device_model: sonoff

10:

submeter_of: 1 # s31_08

device_model: sonoff

11:

submeter_of: 1 # s31_09

device_model: sonoff

12:

submeter_of: 3 # s31_10

device_model: sonoff

13:

submeter_of: 3 # s31_11

device_model: sonoff

14:

submeter_of: 1 # z_shelly_a_A

device_model: shelly

15:

submeter_of: 2 # z_shelly_a_B

device_model: shelly

16:

submeter_of: 2 # z_shelly_b_A

device_model: shelly

17:

submeter_of: 3 # z_shelly_b_B

device_model: shelly

```
appliances:
```

```
- {type: computer, instance: 1, meters: [4]}
- {type: flat screen, instance: 1, meters: [5]}
- {type: projector, instance: 1, meters: [6]}
- {type: broadband router, instance: 1, meters: [7]}
- {type: ethernet switch, instance: 1, meters: [8]}
- {type: desktop computer, instance: 1, meters: [9]}
- {type: desktop computer, instance: 2, multiple: true, count:
2, meters: [10]}
- {type: desktop computer, instance: 3, multiple: true, count:
2, meters: [11]}
- {type: desktop computer, instance: 4, multiple: true, count:
2, meters: [12]}
- {type: desktop computer, instance: 5, multiple: true, count:
2, meters: [13]}
- {type: air conditioner, instance: 1, meters: [14, 15]}
- {type: air conditioner, instance: 2, meters: [16, 17]}
```

Una se tiene los archivos de metadata y se tienen los archivos CSV originales, se puede iniciar la conversión del dataset al formato HDF5. Para ello se desarrolló el archivo CLEDUtills.py el cual está próximo a ser enviado a su revisión para que se integre oficialmente al proyecto NILMTK. Dicho archivo tiene el método `convert_dataset()` el cual recibe como parámetros la carpeta donde se colocan los archivos originales CSV y el nombre de salida del archivo. Es indispensable que en el mismo directorio se encuentre la carpeta `CLED/metadata`. A continuación, se adjunta un fragmento de dicho código.

Para que se proceda con la inclusión en el proyecto oficial, este código debe estar correctamente documentado.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```



```

import pandas as pd
from os.path import *
from os import getcwd
from os import listdir
from nilmtk.datastore import Key
from nilmtk.measurement import LEVEL_NAMES
from nilmtk.utils import check_directory_exists, get_datastore, get_module_directory
from nilm_metadata import convert_yaml_to_hdf5
from sys import getfilesystemencoding

# Column name mapping
columnNameMapping = {'P': ('power', 'active'),
                    'V': ('voltage', ''),
                    'I': ('current', '')}

TIMESTAMP_COLUMN_NAME = "unix_ts"
TIMEZONE = "America/Guayaquil"

def convert_cled(input_path, output_filename, format='HDF'):
    """
    Based on AMPDS CONVERTER.
    Convert CLED 2024.Download the files
    as CSVs and put them in the `input_path` folder for conversion.

    Parameters:
    -----
    input_path: str
        The path of the directory where all the csv
        files are supposed to be stored
    output_filename: str
        The path of the h5 file where all the
        standardized data is supposed to go. The path
        should refer to a particular file and not just a
        random directory in order for this to work.
    format: str
        Defaults to HDF5
    """
    check_directory_exists(input_path)
    files = [f for f in listdir(input_path) if isfile(join(input_path, f)) and
            '.csv' in f and '.swp' not in f]
    files.sort()

    assert isdir(input_path)
    store = get_datastore(output_filename, format, mode='w')

```

```

for i, csv_file in enumerate(files):
    key = Key(building=1, meter=(i + 1))
    print('Loading file #', (i + 1), ' : ', csv_file, '. Please wait...')
    df = pd.read_csv(join(input_path, csv_file))
    df.columns = [x.replace(" ", "") for x in df.columns]
    df.index = pd.to_datetime(df[TIMESTAMP_COLUMN_NAME], unit='s', utc=True)
    df = df.drop(TIMESTAMP_COLUMN_NAME, 1)
    df = df.tz_convert(TIMEZONE)
    df.columns = pd.MultiIndex.from_tuples(
        [columnNameMapping[x] for x in df.columns],
        names=LEVEL_NAMES
    )
    df = df.apply(pd.to_numeric, errors='ignore')
    df = df.dropna()
    df = df.astype(np.float32)
    store.put(str(key), df)
    print("Done with file #", (i + 1))

store.close()
print('Processing metadata...')
convert_yaml_to_hdf5('./CLEd/metadata', output_filename)

```

Sobre este dataset se elaboró un artículo científico llamado “*CLED: Computer Lab Energy Dataset*” el cual fue aprobado para ser mostrado y publicado en *2024 IEEE International Symposium on Measurements and Networking*, lo cual avala la importancia de este resultado obtenido al crear un dataset de energía eléctrica focalizado en Ecuador.

4.2 Estados de operación por dispositivo

A continuación, se presentan los resultados de cada dispositivo descrito en la tabla anterior junto con un análisis inferencial de la causa de ciertos resultados.

Para definir los estados finales de cada dispositivo, aplicaremos una media aritmética entre los distintos métodos aplicados sobre un mismo valor de K . Esto bajo la siguiente fórmula.

$$P_{O_k} = \frac{P_{kmeans} + P_{GMM} + P_{KDE}}{3} \quad (7)$$

Donde:

P_{O_k} : Es el estado de operación al que hacemos referencia.

P_{kmeans} : Es la media para ese clúster obtenida por Kmeans.

P_{GMM} : Es la media para ese clúster obtenida por GMM

P_{KDE} : Es la media para ese clúster obtenida por KDE.

4.2.1 Proyector

El análisis de operación de este dispositivo ya fue realizado como ejemplo para describir la metodología a seguir para cada dispositivo en la sección anterior.

Tabla 17. Proyector Kmeans

| K | Cluster | Medias |
|---|---------|--------|
| 3 | 1 | 5.93 |
| | 2 | 89.68 |
| | 3 | 246.16 |
| 4 | 1 | 5.93 |
| | 2 | 88.81 |
| | 3 | 204.1 |
| | 4 | 248.6 |

Tabla 18. Proyector GMM

| K | Cluster | Medias | Covarianzas | Pesos |
|---|---------|--------|-------------|--------|
| 3 | 1 | 5.76 | 0.2 | 0.12 |
| | 2 | 87.71 | 294.8 | 0.06 |
| | 3 | 246.19 | 112.1 | 0.81 |
| 4 | 1 | 5.77 | 0.21 | 0.12 |
| | 2 | 88.03 | 0.11 | 0.059 |
| | 3 | 217.58 | 1556.1 | 0.1243 |
| | 4 | 249.58 | 0.79 | 0.69 |

Tabla 19. Proyector KDE

| Cluster | Muestras | Media | Mediana | Std | Max | Min | Pesos |
|---------|----------|--------|---------|-------|-----|-----|--------|
| 1 | 1709 | 5.93 | 6 | 1.49 | 24 | 1 | 0.12 |
| 2 | 880 | 88.43 | 99 | 3.07 | 120 | 61 | 0.063 |
| 3 | 18 | 146.61 | 148.5 | 9.28 | 162 | 131 | 0.0012 |
| 4 | 11271 | 246.15 | 249 | 10.71 | 253 | 167 | 0.812 |

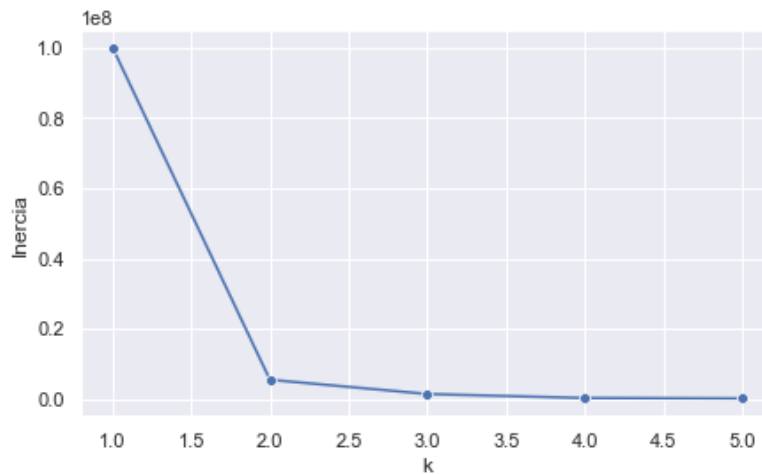


Ilustración 24. Inercia para el dispositivo proyector

Del método de inercia podemos adelantar que un valor óptimo para el número de clústeres será 3.

De Kmeans y GMM observamos que los estados repetidos están entre 5.93, 89 y 247. Tenemos un último estado de operación cuando $K = 4$ entre 204 y 217, sin embargo, por la información de los pesos vemos que este solo representa el 12% del total. Esto lo convertiría en un estado relevante, sin embargo, la desviación estándar que presenta es muy grande comparada con el resto de los estados, esto lo convertiría en un punto aberrante del clúster centrado en 246.

Del análisis de KDE, observamos que el clúster 3 centrado en 146 representa menos del 1%, esta puede ser una generalización de todos los valores encontrados entre los clústeres 2 y 3 de los otros métodos. Por eso la descartamos para este análisis.

El set final de potencias de operación para este dispositivo es el siguiente:

Tabla 20. Estados de operación del Proyector

| Estado | Potencia[W] | Descripción |
|--------|-------------|------------------------|
| 1 | 5.87 | Modo apagado |
| 2 | 88.6 | Modo Ahorro de energía |
| 3 | 246.98 | Modo operación normal |

4.2.2 Monitor

Este dispositivo es uno muy común en los datasets para desagregación de energía y es bien conocido por ser un dispositivo sin múltiples estados de operación, es decir, funciona como un Switch con solo dos estados ON y OFF.

Esto va a depender de la tecnología de los monitores, puesto que monitores antiguos usaban tubos de rayos catódicos, la misma tecnología que televisores en la década de los 90s. En la actualidad, las tecnologías más populares para monitores incluyen LCD, LED y OLED. Cada tecnología tiene un consumo de energía variable. Los monitores LCD y LED suelen ser eficientes en energía, mientras que los OLED pueden ser eficientes en áreas de pantalla con contenido negro. Sin embargo, independiente de la tecnología se coincide que es un dispositivo de bajo consumo de energía, siempre y cuando generalicemos monitores de oficina que no presenten características o funcionalidades demandantes como altas tasas de refresco o resoluciones muy grandes.

De los métodos para clustering obtenemos los siguientes resultados:

Tabla 21. Monitor Kmeans

| K | Cluster | Medias |
|---|---------|--------|
| 3 | 1 | 10.96 |
| | 2 | 12 |
| | 3 | 13 |
| 4 | 1 | 3.14 |
| | 2 | 10.99 |
| | 3 | 12 |
| | 4 | 13 |

Tabla 22. Monitor GMM

| K | Cluster | Medias | Covarianzas | Pesos |
|---|---------|--------|-------------|-------|
| 3 | 1 | 10.96 | 0.25 | 0.4 |
| | 2 | 12 | 0 | 0.39 |
| | 3 | 13 | 0 | 0.2 |
| 4 | 1 | 3.49 | 5 | 0.001 |
| | 2 | 11 | 0 | 0.4 |
| | 3 | 12 | 0 | 0.39 |
| | 4 | 13 | 0 | 0.2 |

Método: KDE

Tabla 23. Monitor KDE

| Cluster | Muestras | Media | Mediana | Std | Max | Min | Peso |
|---------|----------|-------|---------|------|-----|-----|-------|
| 1 | 147 | 2.96 | 4 | 1.59 | 6 | 1 | 0.001 |
| 2 | 95078 | 11.79 | 12 | 0.75 | 14 | 7 | 0.999 |

Inercia:

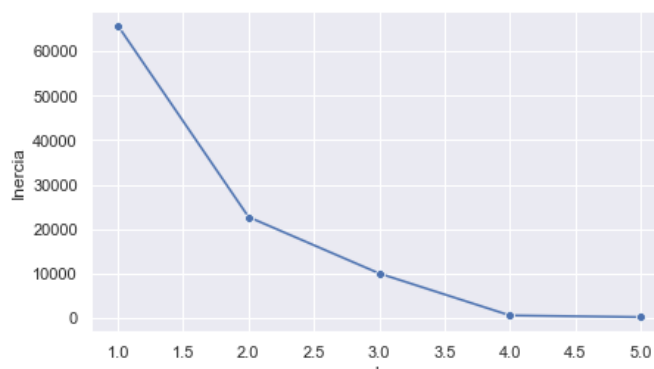


Ilustración 25. Inercia para dispositivo Monitor

Usando el gráfico de inercia podríamos empezar con el número de clústeres 3 o 4, lo cual si analizamos los resultados de los métodos es coherente con este resultado. En todos los métodos vemos clústeres casi centrados en 11, 12 y 13 lo cual podría considerarse como modos de operación. Sin embargo, cuando analizamos la literatura y la opinión de expertos, el contexto de la operación eléctrica total también influye, porque esta variación de 1 Watt entre pseudo-estados no representa una variación sustancial en el total de la potencia consumida en un instante de tiempo. En contexto, tenemos equipos que consumen Watts en

el orden de 10^2 e incluso sus modos de operación varían en torno a dos cifras de magnitud. Es por ello que una variación tan pequeña no se la considera como un estado de un conjunto de multi-estados, sino que forman parte del mismo estado ON como lo había descrito anteriormente.

Es así que partimos de la solución de KDE como la mejor representación, porque nos indica que la varianza está en termino aceptables y adicional, ese centroide es bastante acertado con los otros métodos si es que consideramos unir los estados alrededor de los 12W.

Se identifica un clúster que representa menos del 1% cerca del 0 (2.46) pero este clúster tiene un peso tan bajo que lo asignaremos al estado OFF.

La tabla de estados queda de la siguiente forma:

Tabla 24. Estados de operación del Monitor

| Estado | Potencia |
|--------|----------|
| OFF | 0 |
| ON | 11.8 |

4.2.3 Router

Entre los dispositivos comúnmente usados se encuentra un router inalámbrico en modo Access point de la marca DLINK modelo DIR-825 el cual se considera en la gama de entrada. Este AP es usado principalmente para distribuir la conexión inalámbrica dentro del laboratorio, donde usualmente atiende hasta 15 conexiones simultaneas, entre dispositivos personales como teléfonos y laptops, hasta dispositivos IoT.

Según la literatura, los routers inalámbricos dependiendo del modelo consumen de 2 a 20 Watts de potencia, según esta tabla:

Tabla 25. Modelos de Router comunes

| Modelo de router: | Potencia (W): |
|--------------------------|----------------------|
| TRENDnet TEW-651BR | 2 W |
| Asus ZenWiFi AX (XT8) | 15 W |
| Netgear Orbi Pro | 10.5 W |
| Asus RT-AX86U | 10 W |

| | |
|----------------------|-----|
| TP-Link Deco M9 Plus | 6 W |
|----------------------|-----|

Mediante una inspección visual con histograma de los valores de potencia podemos encontrar que los datos están mayoritariamente en los 6W, algo que coincide con el router inalámbrico mencionado en la tabla anterior, puesto que se encuentran en la misma categoría de uso.

Mediante un resumen estadístico los valores para este dispositivo son como media 5.92 con una desviación estándar de 0.34, lo cual podemos corroborar en el siguiente gráfico de frecuencias.

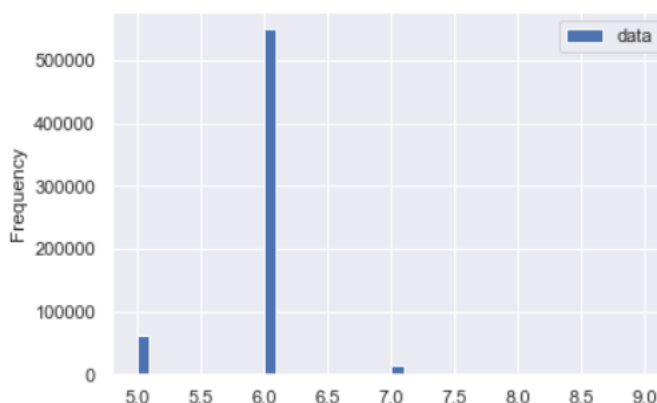


Ilustración 26. Gráfico de frecuencias para el dispositivo Router

Se puede determinar que el 95% de las muestras corresponden a un valor de potencia muy cercano a 6[W], lo cual justifica el no realizar el resto de las técnicas de clustering. Como resultado final se obtiene la siguiente tabla de operación.

Tabla 26. Estados de operacion del router

| Estado | Potencia |
|--------|----------|
| OFF | 0 |
| ON | 6 |

4.2.4 Switch

Este dispositivo se encuentra en operación junto con el dispositivo anterior, para satisfacer la necesidad de conexión entre distintos dispositivos y estaciones de comunicación. Este dispositivo es de la marca DLINK, y funciona con un encendido

permanente. Según la literatura y por la experiencia este dispositivo tiene un funcionamiento similar al Router, lo cual se comprueba en el siguiente histograma.

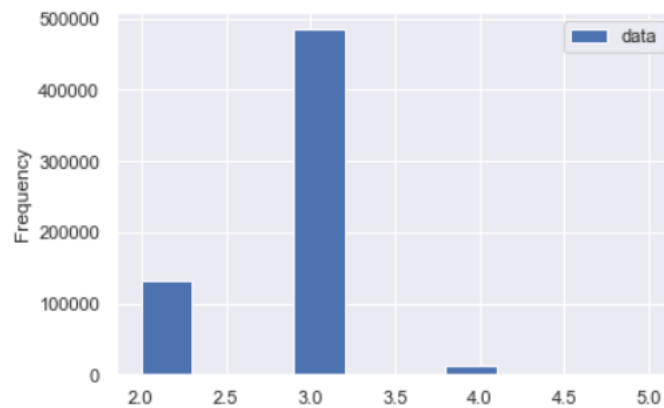


Ilustración 27. Gráfico de frecuencias para dispositivo Switch

Se puede ver claramente dos clústeres, uno centrado en 3 y otro centrado en 2, sin embargo, los estadísticos muestran que la media de la serie es 2.81 con una desviación estándar de 0.44, lo cual por el orden de magnitud que de estos valores se desestima el encontrar clústeres con otra técnica.

Los estados de operación se definen en la siguiente tabla:

Tabla 27. Estados de operación del switch

| Estado | Potencia |
|--------|----------|
| OFF | 0 |
| ON | 3 |

4.2.5 PC's

Según la literatura, el consumo de energía de una computadora normal puede variar significativamente según su hardware, uso y eficiencia energética. En general, una computadora de escritorio típica consume entre 200 y 800 vatios en funcionamiento normal. Esto incluye la energía consumida por la unidad central de procesamiento (CPU), la tarjeta gráfica, la placa madre, el disco duro, la memoria RAM y otros componentes. Las laptops y computadoras portátiles suelen ser mucho más eficientes en términos de energía, consumiendo entre 20 y 100 vatios en promedio. Sin embargo, este consumo puede aumentar significativamente cuando la computadora realiza tareas más intensivas, como juegos o edición de video.

Esta variabilidad, nos dirigió a no tomar solo 1 PC como referencia, sino tener varios modelos (PC0, PC1, PC2, PC3, PC4) para ampliar nuestro espectro del estudio y obtener un perfil de carga que represente significativamente a varios modos de operación y equipos.

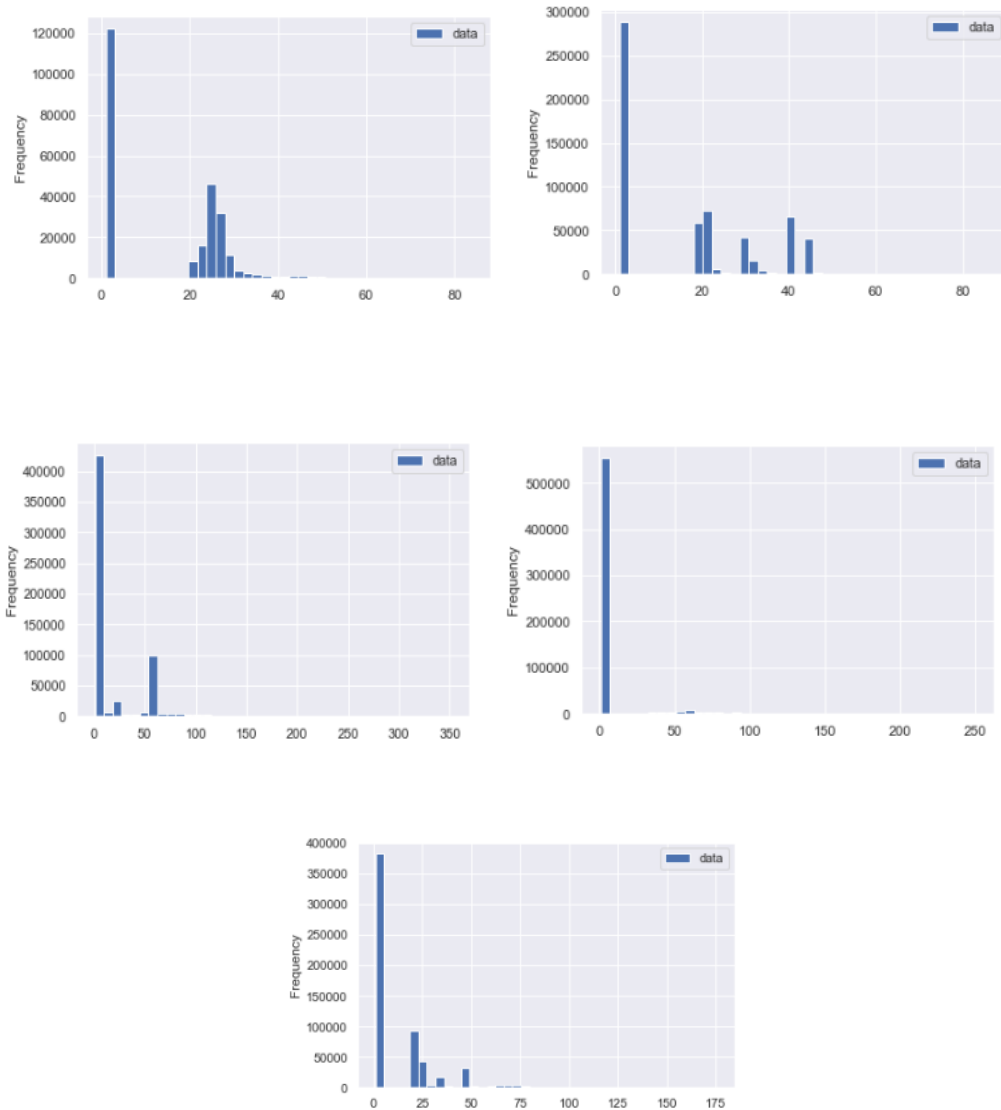


Ilustración 28. Gráficos de frecuencia para PC0 a PC4

De los gráficos anteriores podemos visualizar que todos los dispositivos de prueba tienen un consumo significativo cerca de los cero vatios, pero que es distinto a cero, dado que cuando una computadora está completamente apagada, su consumo de energía es mínimo, pero no es igual a cero. La mayoría de las computadoras

modernas, incluso cuando están apagadas, consumen una pequeña cantidad de energía para mantener ciertos componentes en espera, como la placa base, la tarjeta de red o la funcionalidad de encendido remoto. Esto se hace para que la computadora pueda encenderse más rápido cuando se presiona el botón de encendido o para permitir que se realicen tareas programadas, como actualizaciones del sistema. El consumo de energía en modo apagado, también conocido como "modo de espera" o "standby" varía según el modelo de la computadora, pero suele estar en el rango de 1 a 5 vatios en la mayoría de las computadoras modernas.

A diferencia de los dispositivos evaluados anteriormente, este consumo no se lo puede despreciar ni acercarse exactamente a cero, dado que, por la naturaleza del dispositivo, el hábito de consumo hace que este sea un estado que constantemente se visualice en la operación normal de un sistema. Para evaluar los posibles clústeres aplicaremos el método explicado a inicios del capítulo.

Tabla 28. PC0 Kmeans

| Kmeans | | |
|--------|---------|--------|
| K | Cluster | Medias |
| 3 | 1 | 1 |
| | 2 | 25.35 |
| | 3 | 43.46 |

Tabla 29. PC0 GMM

| GMM | | | | |
|-----|---------|--------|-------------|-------|
| K | Cluster | Medias | Covarianzas | Pesos |
| 3 | 1 | 1 | 0.001 | 0.48 |
| | 2 | 25 | 4.27 | 0.43 |
| | 3 | 35.7 | 10.1 | 0.07 |

Tabla 30. PC0 KDE

| KDE | | | | | | | |
|---------|----------|-------|---------|------|-----|-----|------|
| Cluster | Muestras | Media | Mediana | Std | Max | Min | Peso |
| 1 | 122102 | 1 | 1 | 0.04 | 2 | 1 | 0.48 |
| 2 | 81 | 8.41 | 8 | 2.57 | 15 | 4 | 0.01 |
| 3 | 130333 | 26.67 | 25 | 5.7 | 84 | 18 | 0.51 |

Tabla 31. PC1 Kmeans

| Kmeans | | |
|--------|---------|--------|
| K | Cluster | Medias |
| 3 | 1 | 1.48 |
| | 2 | 23.2 |
| | 3 | 41.98 |

Tabla 32. PC1 GMM

| GMM | | | | |
|-----|---------|--------|-------------|-------|
| K | Cluster | Medias | Covarianzas | Pesos |
| 3 | 1 | 1.48 | 0.25 | 0.47 |
| | 2 | 23.67 | 29.38 | 0.338 |
| | 3 | 42.25 | 2.73 | 0.18 |

Tabla 33. PC1 KDE

| KDE | | | | | | | |
|---------|----------|-------|---------|------|-----|-----|------|
| Cluster | Muestras | Media | Mediana | Std | Max | Min | Peso |
| 1 | 288322 | 1.48 | 1 | 0.5 | 11 | 1 | 0.47 |
| 2 | 139178 | 20.33 | 21 | 1.43 | 26 | 12 | 0.23 |
| 3 | 173363 | 38.01 | 41 | 6.1 | 86 | 27 | 0.28 |

Tabla 34. PC2 Kmeans

| Kmeans | | |
|--------|---------|--------|
| K | Cluster | Medias |
| 3 | 1 | 3.86 |
| | 2 | 60.92 |
| | 3 | 193.4 |

Tabla 35. PC2 GMM

| GMM | | | | |
|-----|---------|----------|-------------|-------|
| K | Cluster | Medias | Covarianzas | Pesos |
| 3 | 1 | 2.57 | 0.27 | 0.701 |
| | 2 | 50.13 | 339 | 0.263 |
| | 3 | 142.0026 | 470 | 0.027 |

Tabla 36. PC2 KDE

| KDE | | | | | | | |
|---------|----------|-------|---------|------|-----|-----|-------|
| Cluster | Muestras | Media | Mediana | Std | Max | Min | Peso |
| 1 | 425809 | 2.54 | 3 | 0.5 | 9 | 1 | 0.71 |
| 2 | 40136 | 21.99 | 21 | 7.29 | 46 | 10 | 0.066 |
| 3 | 134575 | 70.11 | 57 | 37.1 | 352 | 47 | 0.22 |

Tabla 37. PC3 Kmeans

| Kmeans | | |
|--------|---------|--------|
| K | Cluster | Medias |
| 3 | 1 | 1.9 |
| | 2 | 53.59 |
| | 3 | 89.05 |

Tabla 38. PC3 GMM

| GMM | | | | |
|-----|---------|--------|-------------|-------|
| K | Cluster | Medias | Covarianzas | Pesos |
| 3 | 1 | 1.83 | 0.53 | 0.921 |
| | 2 | 53.98 | 288 | 0.05 |
| | 3 | 82.155 | 544 | 0.023 |

Tabla 39. PC3 KDE

| KDE | | | | | | | |
|---------|----------|-------|---------|-------|-----|-----|--------|
| Cluster | Muestras | Media | Mediana | Std | Max | Min | Peso |
| 1 | 553622 | 1.83 | 2 | 0.73 | 14 | 1 | 0.921 |
| 2 | 2389 | 18.31 | 17 | 3.14 | 28 | 15 | 0.0001 |
| 3 | 44716 | 64.94 | 61 | 20.97 | 250 | 29 | 0.0744 |

Tabla 40. PC4 Kmeans

| Kmeans | | |
|--------|---------|--------|
| K | Cluster | Medias |
| 3 | 1 | 2.7 |
| | 2 | 23.26 |
| | 3 | 57.75 |

Tabla 41. PC4 GMM

| GMM | | | | |
|-----|---------|--------|-------------|-------|
| K | Cluster | Medias | Covarianzas | Pesos |
| 3 | 1 | 2.72 | 0.2 | 0.63 |
| | 2 | 21.42 | 1.75 | 0.214 |
| | 3 | 48.53 | 39 | 0.14 |

Tabla 42. PC4 KDE

| KDE | | | | | | | |
|---------|----------|-------|---------|-------|-----|-----|-------|
| Cluster | Muestras | Media | Mediana | Std | Max | Min | Peso |
| 1 | 382637 | 2.72 | 3 | 0.44 | 9 | 1 | 0.63 |
| 2 | 135939 | 21.51 | 21 | 1.46 | 28 | 12 | 0.226 |
| 3 | 82152 | 50.77 | 45 | 19.03 | 176 | 29 | 0.136 |

Realizando una interpretación de los resultados, concluimos que todos los dispositivos (evaluando los tres métodos) incluyen un estado de operación definido como “stand-by” alrededor de 2 vatios. Existe un estado común más encontrado en PC0, PC1 y PC4 con suficiente número de muestras para considerarlo representativo y es cerca de los 25 vatios.

Por último, también se identifica un clúster que comprende entre los 50 a 80 vatios, esto puede caracterizarse como un estado de operación especial, tal vez asociado a un proceso en específico de las computadoras que podemos establecer como estado recurrente. El resumen final de los estados se refleja en la siguiente tabla:

Tabla 43. Estados de operación de las PC

| Estado | Potencia |
|----------|----------|
| Stand-by | 2 |
| Estado 1 | 25 |
| Estado 2 | 65 |

4.2.6 Aire acondicionado

Los aires acondicionados (AC) son conocidos por consumir una gran cantidad de electricidad debido a varios factores relacionados con su funcionamiento y diseño. El proceso de refrigeración implica varias etapas como la compresión,

condensación y evaporación de un refrigerante. Este proceso requiere una gran cantidad de energía para el compresor, que es el componente que más energía consume en el sistema. Para enfriar grandes volúmenes de aire o para enfriar espacios grandes, el compresor debe trabajar más intensamente, aumentando el consumo de electricidad. Todo esto se relaciona con la capacidad del aire acondicionado, medida en BTU (British Thermal Units) o en toneladas. Un sistema de mayor capacidad para enfriar áreas más grandes o para mantener temperaturas más bajas consumirá más electricidad.

Los aires acondicionados de este estudio son marca PANASONIC de 24000 BTU y según los manuales oficiales deberían consumir entre 1 kW y 3kW de potencia activa.

Mediante una inspección visual en el siguiente gráfico podemos encontrar el gráfico de frecuencias, donde claramente se visualizan 3 clústeres, dos de ellos visiblemente remarcados alrededor de los 400W y los 1500W, y el ultimo agrupa las mediciones entre 500 y 1300 vatios. Esto se comprueba con las técnicas de clusterización aplicadas que se muestran a continuación.

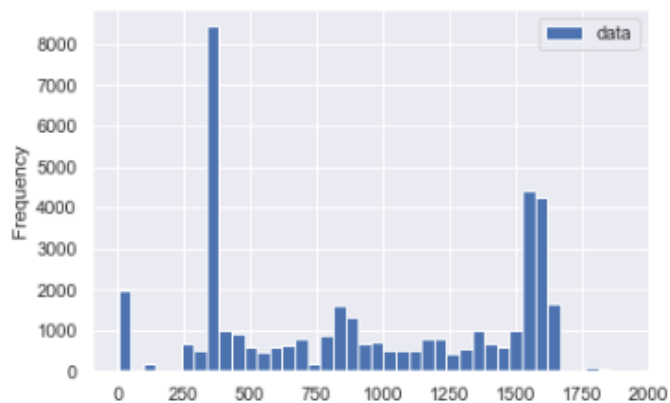


Ilustración 29. Grafico de frecuencias para el Aire Acondicionado

Tabla 44. AC Kmeans

| Kmeans | | |
|--------|---------|--------|
| K | Cluster | Medias |
| 3 | 1 | 331 |
| | 2 | 915.8 |
| | 3 | 1532.8 |

Tabla 45. AC GMM

| GMM | | | | |
|-----|---------|--------|-------------|--------|
| K | Cluster | Medias | Covarianzas | Pesos |
| 3 | 1 | 332 | 2138 | 0.369 |
| | 2 | 1032 | 9359 | 0.363 |
| | 3 | 1580 | 1633 | 0.2673 |

Tabla 46. AC KDE

| KDE | | | | | | | |
|---------|----------|-------|---------|--------|---------|-------|------|
| Cluster | Muestras | Media | Mediana | Std | Max | Min | Peso |
| 1 | 1951 | 24.47 | 34 | 13.8 | 78.7 | 1.5 | 0.05 |
| 2 | 158 | 120.3 | 112.93 | 13.74 | 150.3 | 93.8 | 0.01 |
| 3 | 37612 | 970.6 | 916.11 | 509.94 | 1907.17 | 180.3 | 0.94 |

Tabla 47 Estados de operación AC

| Estado | Potencia |
|-------------|----------|
| OFF | 0 |
| Operación 1 | 330 |
| Operación 2 | 600 |
| Operación 3 | 1500 |

Como pudimos comprobar mediante una inspección visual, tres principales clústeres son identificados por los métodos GMM y Kmeans, sin embargo, KDE no puede caracterizar dichos clústeres. Esto corresponde a los métodos de funcionamiento previamente mencionados, que serían reposo y activación de los compresores. Dado que existe una amplia gama de configuraciones como la velocidad de los ventiladores, aspas y modos de enfriamiento, esto causa que existe un amplio clúster entre los 500W y los 1400W. El clúster con centroide en 330W puede ser un estado con el compresor apagado y los ventiladores circulando a velocidad mínima, mientras que el clúster con centroide en 1500 W representa un funcionamiento con el compresor encendido y los ventiladores en funcionamiento.

El análisis de este dispositivo nos permite sacar conclusiones interesantes en torno al impacto que tienen las distintas condiciones de operación sobre este tipo

de equipos, como la falta de mantenimiento, funcionar con piezas dañadas. Se debería profundizar el estudio teniendo una línea base de control y otro equipo que permita contrastar las mediciones, dado que al ser el equipo que más consume energía puede resultar en importante información para los distintos sistemas de control y manejo de energía.

4.3 Modelo NILM

Uno de los principales resultados de este dataset era poder entrenar efectivamente un modelo que permita realizar la desagregación no intrusiva de cargas eléctricas, para lo cual ya hemos explicado y detallado las fases que intervienen en este proceso.

Con el dataset descrito en la sección 4.1, se realiza la etapa de entrenamiento usando el API de NILMTK. A continuación, se muestra el objeto que describe al proceso de entrenamiento para el framework.

```
experiment1 = {
  'power': {'mains': ['active'],'appliance': ['active']},
  'sample_rate': 10,
  'appliances': ['projector', 'air conditioner', 'desktop computer', 'flat
screen'],
  'methods': {'Mean':Mean(), 'CO': CO(), 'Hart': Hart85(),
              'DAE': DAE(), 'Seq2Point': Seq2Point(), 'RNN': RNN()},
  'train': {
    'datasets': {
      'ULLED': {
        'path': './cled.h5',
        'buildings': {
          1: {
            'start_time': '2024-01-01',
            'end_time': '2024-01-15'
          }
        }
      }
    }
  }
}
```

```
    },
    'test': {
      'datasets': {
        'ULLED': {
          'path': './cled.h5',
          'buildings': {
            1: {
              'start_time': '2024-01-16',
              'end_time': '2024-01-20'
            }
          }
        }
      },
      'metrics':['f1score','nde', 'mae']
    }
  }
}
```

En este objeto se detallan que variables se deben usar para el entrenamiento tanto del lado de los medidores como de los dispositivos, el tiempo de muestreo, los dispositivos seleccionados y los distintos modelos existentes en el framework. Para este estudio se seleccionaron 3 algoritmos comunes de optimización en adición con 3 modelos de redes neuronales presentes en la extensión NILMTK-CONTRIB, estos son *Combinatorial Optimization*, *Hart85* y *Mean* para los algoritmos de optimización, y *Denoising Autoencoder*, *Sequence to Point* y *Recurrent Neural Networks* para el apartado de redes neuronales. Se define la sección de entrenamiento y la sección de testeo, declarando la ruta donde se encuentra nuestro dataset en formato HDF5 , que edificio o lugar va a ser testado, esto referente a la metadata explicada en secciones anteriores, y las fechas exactas de la data a ser tomada por el algoritmo.

Para el siguiente benchmark se decidió tomar las fechas del 1 al 15 de enero de 2024 para entrenar los modelos y del 16 al 20 de enero de 2024 para testear.

En la siguiente grafica se observa el ground truth de un segmento de los datos mostrados. En dicha grafica tenemos las tres fases y los elementos que se

encuentran en cada fase. Esta información fue la detallada en el archivo building.yml de la sección 4.1.1.

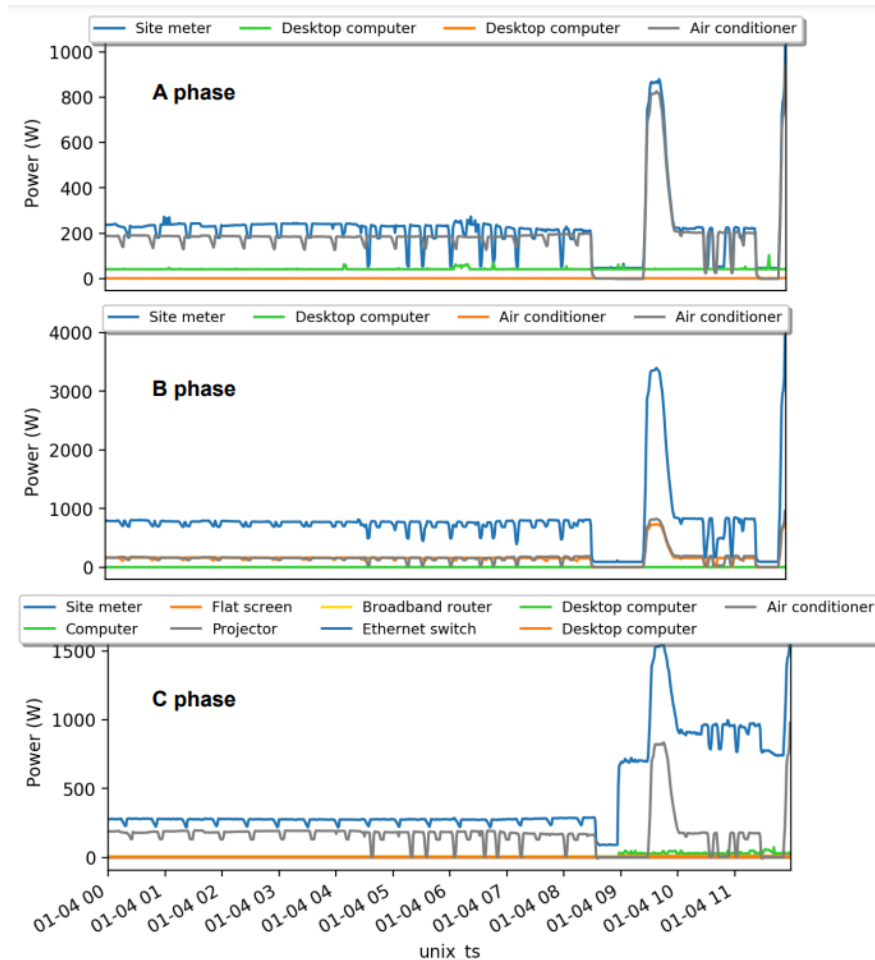


Ilustración 30. Representación gráfica de las señales de los dispositivos y medidores con metadata

Esta es una de las ventajas de trabajar el modelo usando el framework NILMTK, dado que, si quisiéramos trabajarlo con todas las señales agregadas sin la distinción de la metadata, obtendríamos algo como la siguiente gráfica, lo cual justifica el usar un framework para definir la información propia del sistema.

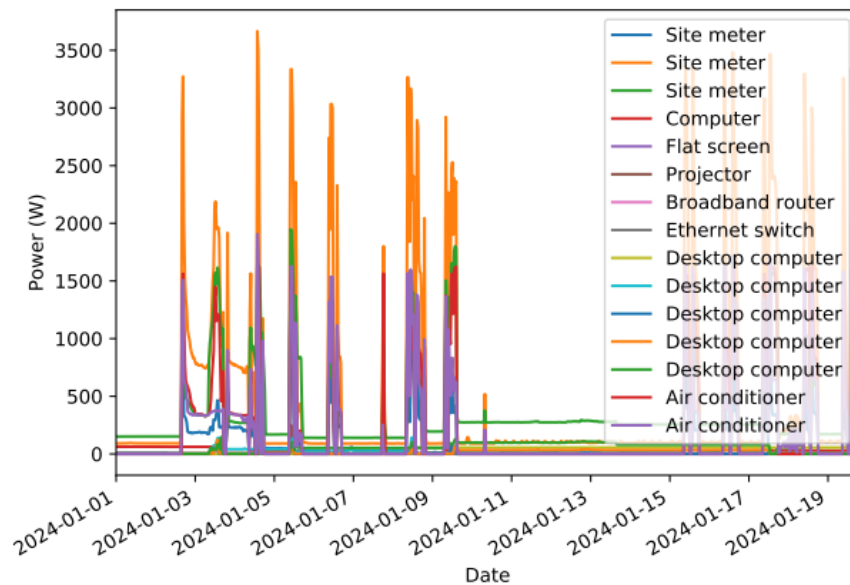


Ilustración 31. Representación de las señales eléctricas sin clasificar por metadato

Una vez definido todas las variables del experimento, habiendo seleccionado los algoritmos y las fechas para el proceso, NILMTK presenta un método conocido como *selección por activaciones*, el cual consiste en filtrar las señales de entrada al entrenamiento para eliminar los periodos de inactividad, es decir solo se ingresa la información que va a aportar al aprendizaje del sistema.

En este tipo de entrenamientos se desea aprender el comportamiento de los dispositivos cuando están encendidos/activados, cuando están apagados o no se están usando esa información es ruido que puede volver poco sensible a nuestro modelo. La mayoría de los trabajos que recopilan el estado del arte de NILM mencionan la importancia de filtrar estos periodos de inactividad. En la siguiente figura se muestra una representación de estos periodos de actividad/inactividad donde los puntos marcados en la línea de tiempo indican que secciones de la ventana de tiempo seleccionada tiene data valiosa para el entrenamiento. Esto se complementa con lo hablado en la sección anterior de los perfiles de carga, ya que estas activaciones pueden ser mejoradas a partir de conocer cuando los dispositivos tienen sus puntos de operación.

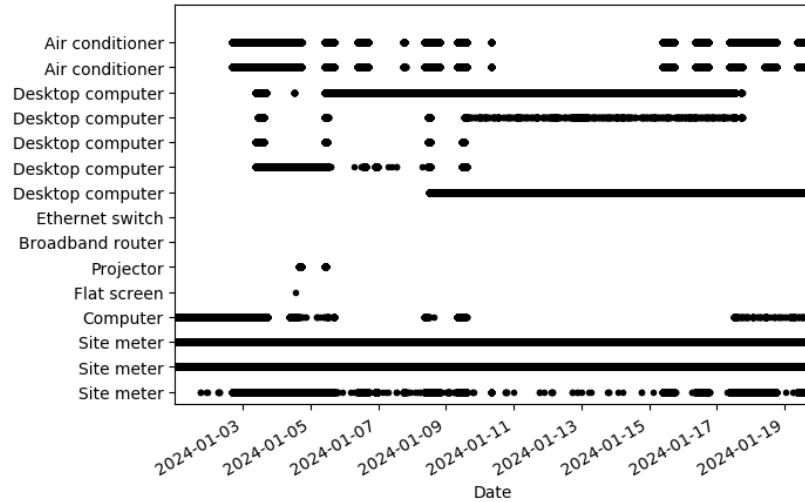


Ilustración 32. Periodos de actividad/inactividad de los dispositivos eléctricos

El experimento aquí mostrado se llevó a cabo con un procesador Intel Core i7 @ 2.90 GHz, 16 GB de memoria RAM y una tarjeta gráfica NVIDIA RTX3060. El tiempo de entrenamiento fueron aproximadamente 33 minutos. En la siguiente tabla se reportan los resultados.

Tabla 48. Métricas de evaluación para los algoritmos de desagregación eléctrica

| Dispositivo | Algoritmo | f1 score | NDE | MAE |
|--------------------|-----------|--------------|--------------|--------------|
| Aire acondicionado | CO | 0.857 | 0.864 | 342.87 |
| | Mean | 0.725 | 0.847 | 298.27 |
| | Hart85 | 0.317 | 0.928 | 305.88 |
| | DAE | 0.861 | 0.389 | 123.44 |
| | Seq2Point | 0.893 | 0.329 | 85.49 |
| | RNN | 0.727 | 0.39 | 128.36 |
| Computadora | CO | 0.703 | 1.2 | 24.117 |
| | Mean | 0 | 0.825 | 16.51 |
| | Hart85 | 0.013 | 1.22 | 19.611 |
| | DAE | 0.463 | 0.791 | 16.853 |
| | Seq2Point | 0.667 | 0.68 | 11.29 |
| | RNN | 0.519 | 0.776 | 15.662 |
| Proyector | CO | 0.124 | 6.84 | 440.7 |
| | Mean | 0 | 0.99 | 20.37 |
| | Hart85 | 0.1 | 0.97 | 18.79 |
| | DAE | 0 | 1.001 | 22.091 |
| | Seq2Point | 0 | 1.001 | 21.52 |
| | RNN | 0 | 1.05 | 23.48 |

| | | | | |
|---------|-----------|--------------|--------------|-------------|
| Monitor | CO | 0.352 | 1.64 | 7.9 |
| | Mean | 0 | 0.945 | 3.339 |
| | Hart85 | 0 | 3.233 | 4.24 |
| | DAE | 0.185 | 1.278 | 6.19 |
| | Seq2Point | 0.501 | 0.743 | 2.87 |
| | RNN | 0.282 | 0.874 | 3.12 |

Las métricas utilizadas para evaluar los algoritmos propuestos fueron *f1score*, que evalúa la capacidad del modelo para detectar y clasificar diferentes dispositivos eléctricos; el Error de Desagregación Normalizado (NDE), una métrica que mide el error normalizado en la desagregación de energía, útil para evaluar qué tan bien un modelo NILM puede estimar la energía consumida por cada dispositivo individual; y el Error Medio Absoluto (MAE), una métrica simple pero efectiva que calcula el promedio de las diferencias absolutas entre las predicciones y los valores reales.

Los algoritmos basados en redes neuronales tienen un mejor rendimiento tanto en la clasificación individual de energía como en el error de desagregación, lo que motiva el estudio y desarrollo de esta área en función de la cantidad de datos. Específicamente, el algoritmo Seq2Point presenta un rendimiento constante en la mayoría de los dispositivos, excepto en el caso del proyector, donde la cantidad de activaciones encontradas en el periodo seleccionado pudo no haber sido suficientemente representativa para que el modelo caracterice sus puntos de operación.

4.4 Modelo de pronóstico para la facturación de consumo eléctrico

En esta sección se reportan los resultados del modelo de pronóstico, para el cual necesitamos definir las siguientes restricciones de la explicación en el capítulo 3.4:

1. Los datos disponibles son n días previos al momento de realizar la predicción.
2. Los datos consisten en las potencias instantáneas por fase.
3. El consumo instantáneo debe traducirse en energía eléctrica consumida en kilovatios-hora (kWh).

4. Asumimos que el máximo número de días a pronosticar será 30 días, 31 dependiendo del mes, empezando el primer día de cada mes.

Dado que el modelo seleccionado para esta tarea, *Chronos*, se basa en una arquitectura de modelos de lenguaje, los pronósticos probabilísticos se obtienen al muestrear múltiples trayectorias futuras dadas en el contexto histórico. Para ello primero debemos evaluar que tan amplio debe ser este contexto.

Considerando que nuestro máximo periodo de evaluación son 30 días, evaluaremos enviar un contexto máximo de ese tamaño. Para ello usaremos las métricas de Error Medio Absoluto (MAE), Error Cuadrático Medio (MSE) y la Raíz del Error Cuadrático Medio (RMSE), dado que son métricas ampliamente utilizadas y valiosas para evaluar la precisión de los modelos de previsión del consumo eléctrico.

A continuación, se muestran 2 meses tomados al azar, donde graficamos los pronósticos realizados a partir de distintos tamaños de contexto y sus respectivas métricas. Los distintos pronósticos se pueden identificar a partir de una escala de color que va desde 1 a 30 días.

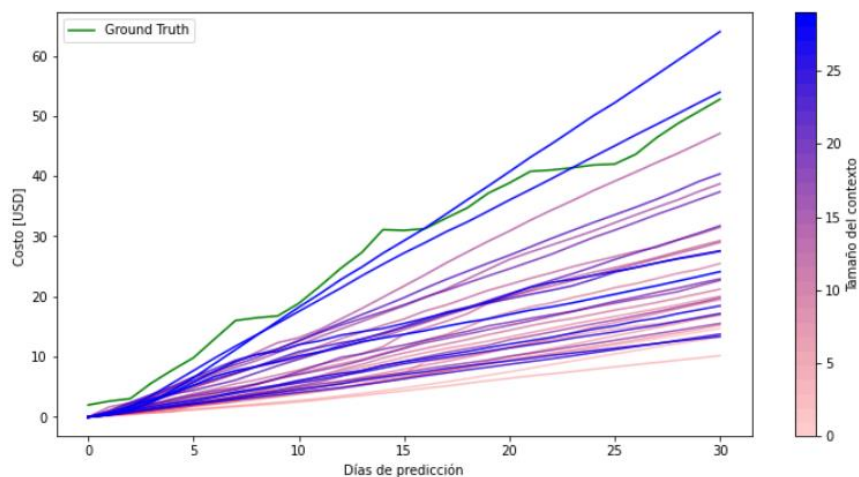


Ilustración 33. Predicciones de consumo, variación de acuerdo al tamaño del contexto, diciembre 2023

Tabla 49. Métricas de error para la variación del tamaño de contexto del mes diciembre

2023

| Tamaño del contexto | MAE | MSE | RMSE |
|----------------------------|------------|------------|-------------|
| 1 | 21.204995 | 568.710937 | 23.847661 |
| 2 | 22.537222 | 638.269547 | 25.263997 |
| 3 | 23.482525 | 701.559329 | 26.486965 |
| 4 | 19.973601 | 499.848906 | 22.357301 |
| 5 | 20.816518 | 539.792243 | 23.233429 |
| 6 | 21.185419 | 565.727492 | 23.785027 |
| 7 | 18.323031 | 422.647815 | 20.5584 |
| 8 | 19.571985 | 473.853271 | 21.768171 |
| 9 | 15.933788 | 310.578275 | 17.623231 |
| 10 | 17.911398 | 396.487827 | 19.912002 |
| 11 | 14.094274 | 239.135943 | 15.464021 |
| 12 | 12.141178 | 181.372144 | 13.467448 |
| 13 | 13.459779 | 226.403238 | 15.046702 |
| 14 | 5.633054 | 38.255626 | 6.185113 |
| 15 | 9.207525 | 100.385198 | 10.019241 |
| 16 | 18.916936 | 444.464774 | 21.082333 |
| 17 | 20.593927 | 540.045207 | 23.238873 |
| 18 | 16.922621 | 360.361573 | 18.983192 |
| 19 | 16.609395 | 346.686225 | 18.619512 |
| 20 | 9.527366 | 111.156104 | 10.54306 |
| 21 | 12.936542 | 203.187701 | 14.254392 |
| 22 | 8.156071 | 79.918416 | 8.93971 |
| 23 | 13.471267 | 232.086217 | 15.234376 |
| 24 | 21.589687 | 594.002667 | 24.37217 |
| 25 | 19.61444 | 489.480974 | 22.124217 |
| 26 | 21.260537 | 579.163101 | 24.065808 |
| 27 | 19.057274 | 461.296454 | 21.477813 |
| 28 | 12.928432 | 217.976577 | 14.76403 |
| 29 | 2.405952 | 6.987411 | 2.643371 |
| 30 | 15.086971 | 298.476666 | 17.276477 |

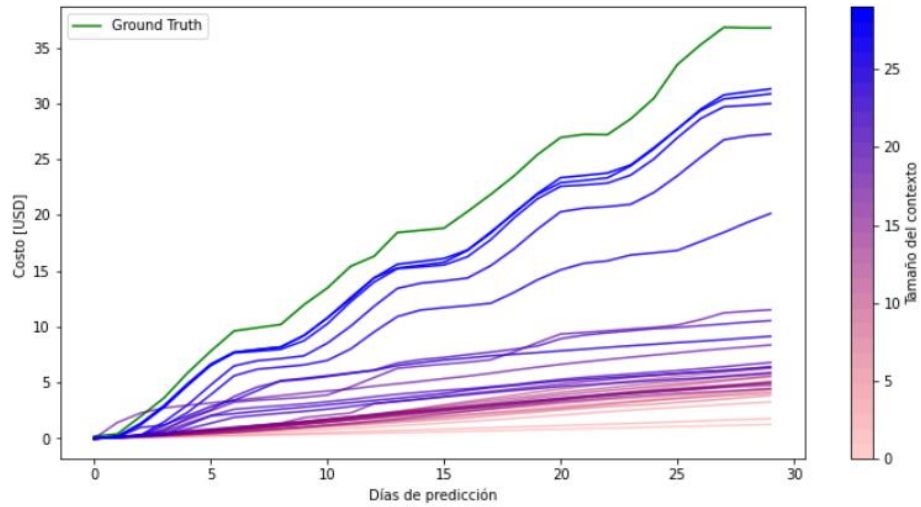


Ilustración 34. Predicciones de consumo, variación de acuerdo al tamaño del contexto, junio 2024

Tabla 50. Métricas de error para la variación del tamaño de contexto del mes junio 2023

| Tamaño del contexto | MAE | MSE | RMSE |
|---------------------|-----------|------------|-----------|
| 1 | 18.509764 | 460.512794 | 21.459562 |
| 2 | 18.276295 | 448.473864 | 21.177202 |
| 3 | 17.639449 | 415.563544 | 20.385376 |
| 4 | 17.096569 | 390.624052 | 19.764211 |
| 5 | 16.614453 | 369.264721 | 19.216262 |
| 6 | 16.514122 | 364.43406 | 19.090156 |
| 7 | 17.33621 | 401.749128 | 20.043681 |
| 8 | 16.900372 | 382.830434 | 19.566053 |
| 9 | 17.179313 | 394.385246 | 19.859135 |
| 10 | 16.679412 | 371.576831 | 19.276328 |
| 11 | 16.10601 | 344.557481 | 18.56226 |
| 12 | 16.269319 | 353.034357 | 18.789209 |
| 13 | 16.41189 | 359.322906 | 18.955815 |
| 14 | 16.683314 | 372.492504 | 19.300065 |
| 15 | 16.721573 | 373.086699 | 19.315452 |
| 16 | 16.951524 | 382.534823 | 19.558497 |
| 17 | 16.552415 | 367.704474 | 19.175622 |
| 18 | 15.773897 | 332.561437 | 18.236267 |
| 19 | 14.008518 | 277.298814 | 16.652292 |
| 20 | 12.907519 | 224.778232 | 14.992606 |
| 21 | 15.178138 | 316.408263 | 17.787868 |
| 22 | 12.685983 | 225.58525 | 15.019496 |

| | | | |
|-----------|-----------|------------|-----------|
| 23 | 15.773515 | 338.11684 | 18.387954 |
| 24 | 15.398716 | 324.139479 | 18.003874 |
| 25 | 13.167936 | 248.188269 | 15.753992 |
| 26 | 8.619982 | 101.855319 | 10.09234 |
| 27 | 5.458302 | 37.487457 | 6.1227 |
| 28 | 3.532134 | 16.422292 | 4.052443 |
| 29 | 3.131828 | 12.866938 | 3.587051 |
| 30 | 3.015028 | 11.781551 | 3.432426 |

En ambos casos alcanzamos niveles de error aceptables cuando tenemos un contexto arriba de 20 días y por en general se obtienen los mejores resultados con 25 a 30 días de contexto.

Todo esto considerando que nuestro modelo de pronóstico es considerado zero-shot, el cual es un tipo de modelo de aprendizaje automático que puede realizar tareas sin haber sido explícitamente entrenado para ellas. En lugar de aprender a partir de ejemplos específicos para cada tarea, un modelo zero-shot utiliza conocimiento previo y contextos generales para inferir cómo resolver nuevas tareas. El preentrenamiento de uno de estos modelos puede mejorar estos resultados, lo cual queda disponible para un trabajo futuro.

4.5 Dashboard de monitoreo de consumo energético y desagregación de cargas.

Uno de los mayores desafíos luego de desarrollar el dataset, los modelos y el sistema de telemetría previamente descrito, era mostrar toda esta información en un dashboard donde se maximice el uso de estos resultados.

Para ello se diseñaron los siguientes componentes:

- a. NilmeC: Aplicación web basada en Angular.
- b. NilmeC-API: Backend para consumo de información eléctrica en tiempo real, histórica y de predicción de consumo.
- c. NilmeC-NILM: Backend para desagregación de consumo energético.

A continuación, se describen cada uno de estos componentes.

4.5.1 NilmeC

Es el aplicativo web desarrollado en Angular que sirve para el consumo de la información. Esta aplicación nace bajo las necesidades expuestas en la sección 3.2.1 e integra la mayoría de los resultados desarrollados en este trabajo, como son las métricas eléctricas, pronóstico de consumo y desagregación de cargas no intrusiva, todo en tiempo real.

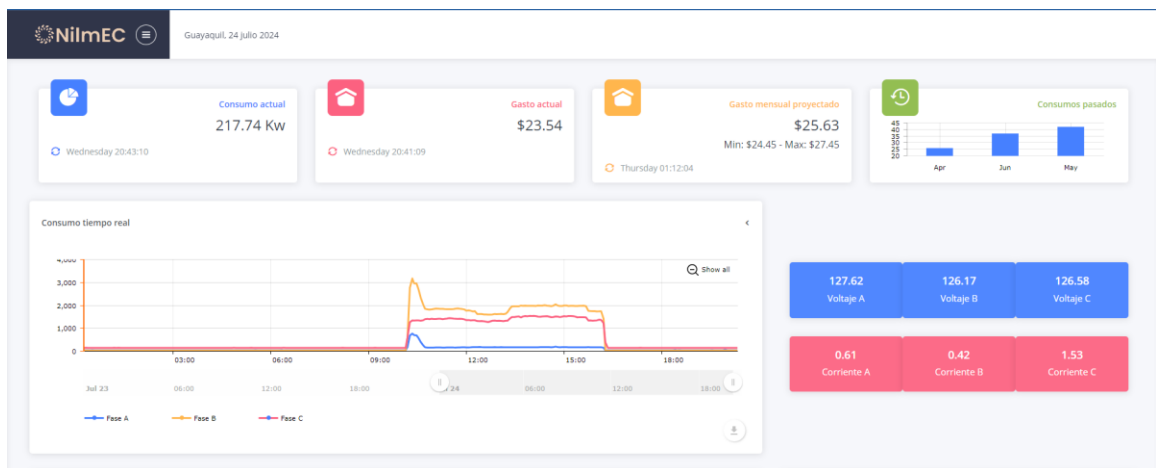


Ilustración 35. NilmeC primera sección

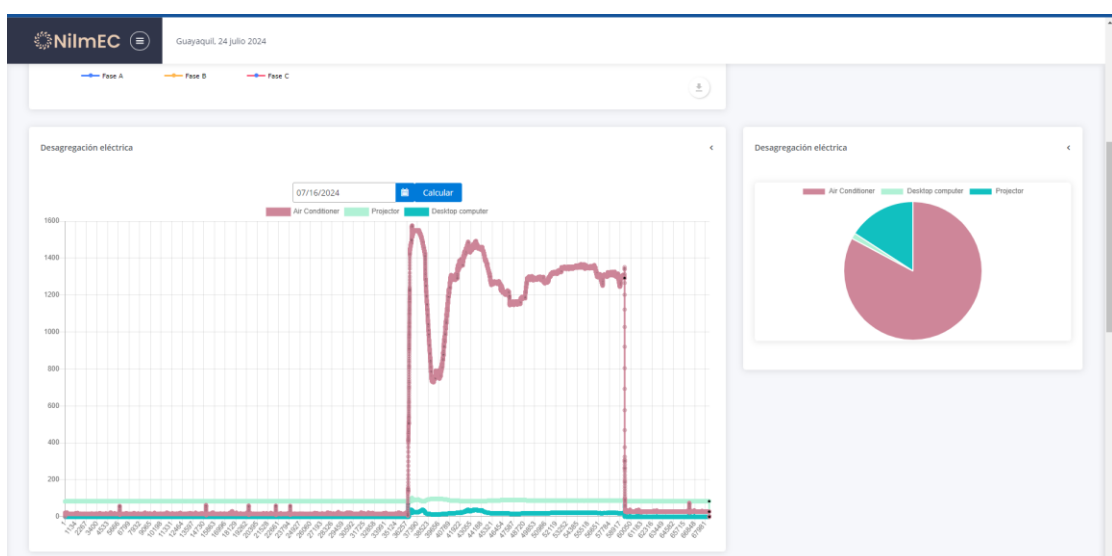


Ilustración 36. NilmeC segunda sección

En la primera sección del aplicativo, Ilustración Ilustración 35 e Ilustración 36 Ilustración, tenemos métricas en tiempo real, como son el consumo instantáneo de la potencia activa registrada por los medidores en el tablero de distribución trifásico, esta tarjeta se actualiza cada 30 segundos y con el monitoreo de este parámetro podemos darnos cuenta de forma rápida cuanta carga tenemos conectada en vivo en el sistema.

El periodo de actualización se eligió porque es un parámetro que varía constantemente dependiendo tanto del modo de operación de los dispositivos como el comportamiento del consumidor, además que el cálculo de este parámetro no significa una gran carga de trabajo para el backend. De igual forma se tiene la tarjeta de Gasto Actual, la cual reporta cuando dinero en USD se ha consumido hasta el momento, esto bajo la metodología explicada en la sección 3.4. A diferencia del gasto actual, esta tarjeta se actualiza cada 5 minutos, debido a que su variabilidad no justifica tener un tiempo de actualización menor y el cálculo de este valor si representa una carga de trabajo para el backend.

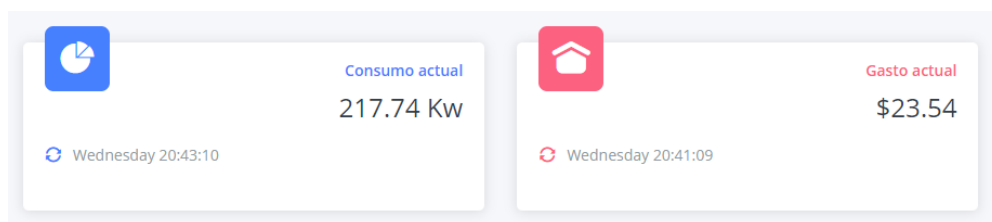


Ilustración 37. Tarjetas de monitoreo: Consumo actual en kW y Gasto actual en USD

En la Ilustración 37 tenemos la tarjeta de Gasto mensual proyectado, el cual se calcula con el contexto (consumo histórico) de los últimos 30 días a partir de la fecha de consulta. El tamaño de este pronóstico será con respecto a los días que faltan para terminar el mes de la consulta. Se actualiza cada 20 minutos, ya que obtener el contexto y realizar la predicción representa una carga significativa para el servidor, calcular una actualización puede tomar de 5 a 8 minutos, es por ello que en el backend esta tarea se calcula programáticamente cada 20 minutos y el

resultado se almacena para luego ser consumido por esta tarjeta de forma asíncrona.

De igual forma la tarjeta de consumos pasados, donde nos muestra la realidad de cuanto se consumió en los tres meses anteriores, dado que este resultado no es necesario actualizarlo constantemente, en el backend se calcula esto cada inicio de mes y se almacena para ser consumido, al cargar el aplicativo se realiza una sola consulta a estos valores.

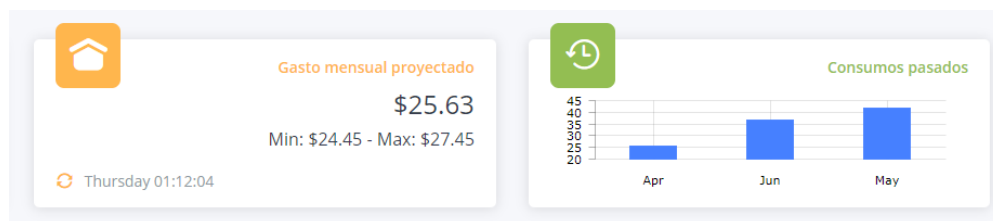


Ilustración 38. Tarjetas de monitoreo: Gasto mensual proyectado y Consumos pasados

En la Ilustración 38 se tiene la tarjeta de monitoreo en tiempo real, donde se muestra el consumo agregado trifásico. Esta información se actualiza cada 20 segundos al igual que la tarjeta de Consumo instantáneo y por defecto nos muestra las últimas 48 horas de consumo. En la parte inferior de un gráfico cuenta con una barra para deslizarnos en la ventana de tiempo y seleccionar un área deseada su visualización. Como se ve en la Ilustración 40, cuenta con opciones para exportación de la data de los gráficos en formatos CSV, XLSX y JSON, así como una función para realizar anotaciones sobre la gráfica e imprimir.

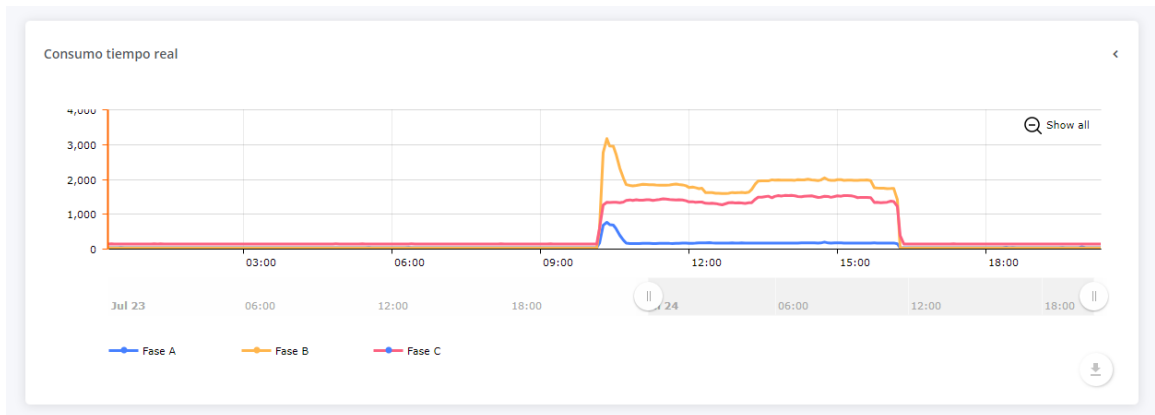


Ilustración 39. Monitoreo en tiempo real, consumo trifásico

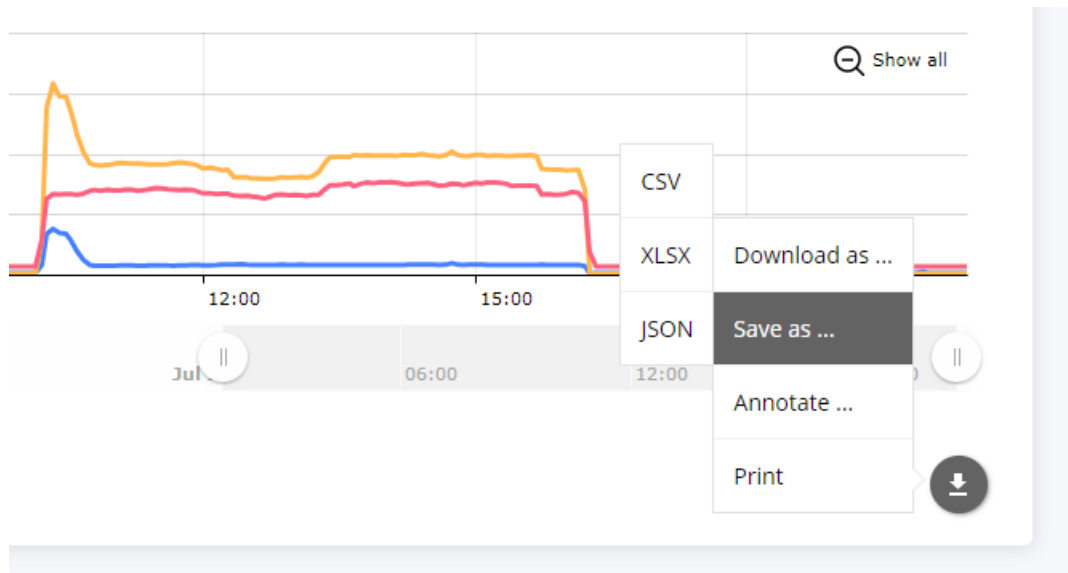


Ilustración 40. Opciones de exportado de datos, anotación e impresión

En la Ilustración 41 se muestran 6 indicadores, que se actualizan cada 20 segundos con los parámetros de operación de la red eléctrica como son los voltajes y corrientes de fase. Esto en conjunto con la tarjeta de Consumo Actual y la gráfica de monitoreo constituyen importantes herramientas para que un administrador del edificio tome decisiones en tiempo real sobre la operación eléctrica.

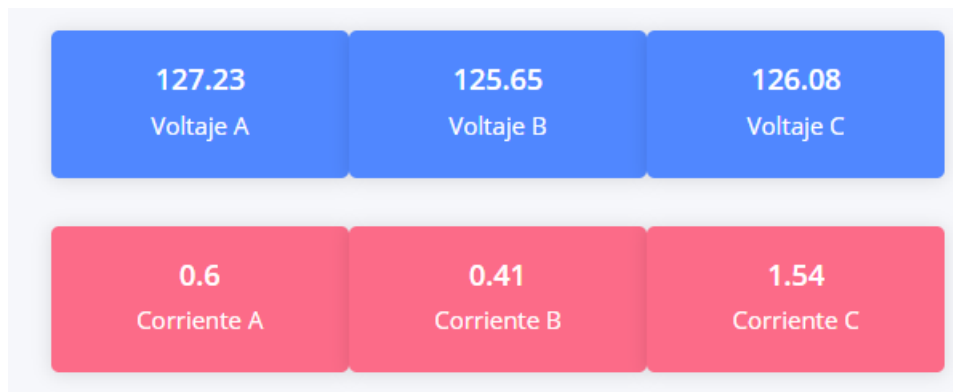


Ilustración 41. Parámetros de operación de la red eléctrica en tiempo real

Sobre la segunda sección encontramos los indicadores para la desagregación eléctrica. En la Ilustración 42 vemos la desagregación por cargas a nivel de consumo eléctrico versus tiempo. Este cálculo se realiza escogiendo un día de acuerdo con el selector mostrado en la Ilustración 43. Por capacidad computacional en esta versión del trabajo se limita a realizar la desagregación sobre máximo un periodo de 24 h.

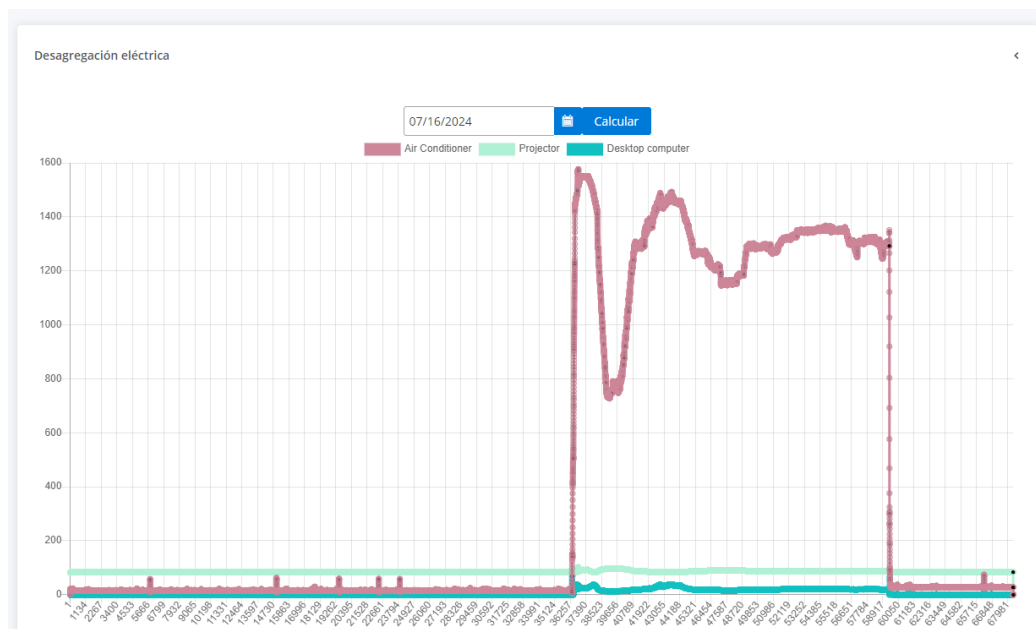


Ilustración 42. Gráfica de desagregación eléctrica no intrusiva.

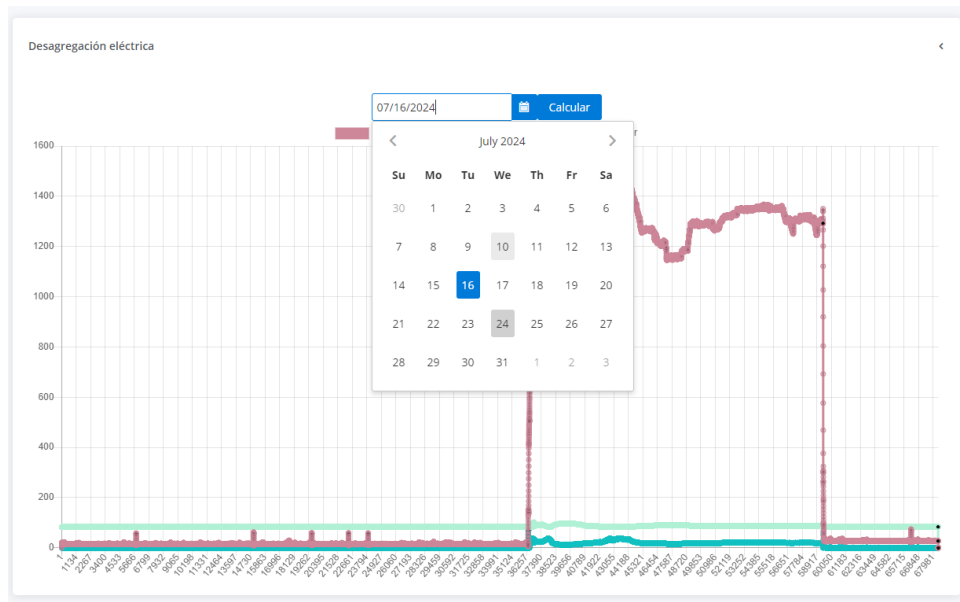


Ilustración 43. Selección de día para la desagregación eléctrica

En la Ilustración 44, tenemos otra forma de visualizar la desagregación del consumo, ya no sobre un eje temporal sino sobre un pie chart que nos permite identificar que porción de la energía está siendo consumida por que dispositivo. Esta es una agregación de toda la energía consumida en el periodo seleccionado.

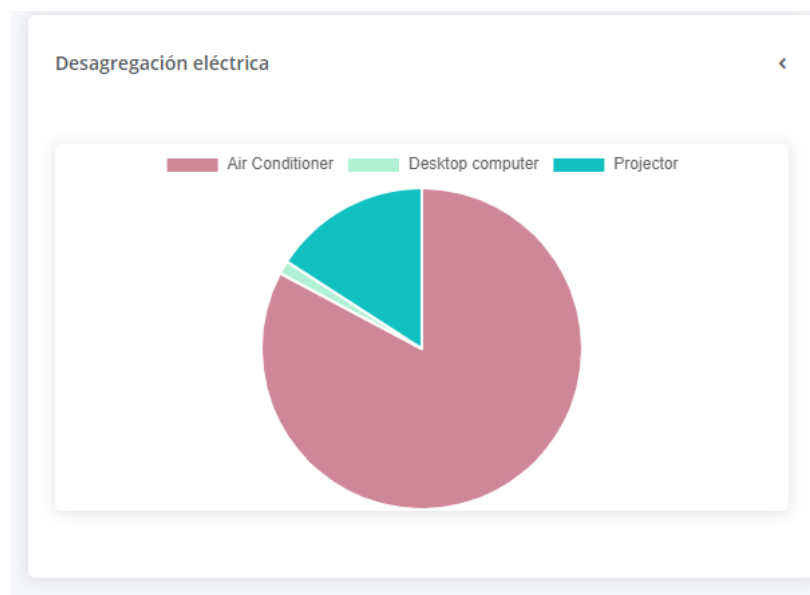


Ilustración 44. Gráfica de porción de energía consumida por dispositivo

4.5.2 NilmeC-API y NilmeC-NILM

Para el consumo de información y punto de inferencia de los modelos tanto de pronóstico como de desagregación se crearon dos backend separados. Ambos escritos en Flask-Python, cuentan con distintos endpoints para aceptar solicitudes HTTP y obscurecer la conexión directamente con la base de datos y el sistema HayloT.

NilmeC-API tiene los métodos para acceder a los consumos en tiempo real, cálculo de la facturación, pronóstico de facturación, consumos históricos y parámetros de operación de la red. En la siguiente tabla se muestran dichos métodos y sus puntos de acceso.

Tabla 51. API Endpoints del backend NilmeC-API

| API Endpoint | Método HTTP | Parámetros/Cuerpo | Definición/Respuesta |
|------------------|-------------|---|---|
| /hayiot/specific | GET | id: identificador del sensor start: fecha inicio consulta end: fecha final de consulta type: variable de consulta | Array con todo el consumo realizado por un dispositivo específico en cierto periodo de tiempo sobre una variable definida |
| /hayiot/groupOf | POST | id: identificador del sensor start: fecha inicio consulta end: fecha final de consulta tags: array con las variables de consulta | Array con todo el consumo realizado por un dispositivo específico en cierto periodo de tiempo sobre un grupo de variables |

| | | | |
|-------------------------------|------|---|--|
| /hayiot/last | POST | id: identificador del sensor tags: array con las variables de consulta | Devuelve la ultima medición registrada por un sensor en especifico sobre un grupo de variables |
| /nilmecapi/getConsumptionBill | GET | - | Devuelve el consumo realizado hasta el momento de la consulta |
| /nilmecapi/getForecast | GET | - | Devuelve en USD el consumo proyectado al final del mes |
| /nilmecapi/getPastMonthsBills | GET | - | Devuelve un array con los consumos realizados los últimos 3 meses en USD |

Debido que el proceso de inferencia de la desagregación eléctrica conlleva varios subprocesos, se decidió separar dicho funcionamiento del backend principal, esto también por temas de mantenimiento de las librerías que usa, dado que al ser un proyecto de código abierto y estar constantemente modificando su contenido con nuevas métricas, algoritmos y datasets puede crear alguna inconsistencia en las librerías de pronóstico. En la siguiente tabla se muestra el único endpoint expuesto.

Tabla 52. Endpoint del backend NilmeEC-NILM

| API Endpoint | Método HTTP | Parámetros/Cuerpo | Definición/Respuesta |
|--------------|-------------|--|---|
| /nilm/ | POST | start: fecha inicio consulta end: fecha final de consulta | Este método devuelve dos objetos, un array con el consumo desagregado por dispositivo y un diccionario con los porcentajes de desagregación por dispositivo |

CAPÍTULO 5

5. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- El uso de técnicas de monitoreo en el campo eléctrico ha revolucionado la forma como los administradores de edificio o sistemas de control de energía toman decisiones, esto motivó el desarrollo de técnicas como el pronóstico de consumos y la desagregación no intrusiva de cargas, lo cual permite optimizar la gestión de estos sistemas eléctricos.
- A pesar de que existen algunos datasets disponibles para investigación en el campo de la energía eléctrica, no se comparan en tamaño ni diversidad a los disponibles para otras áreas de investigación como el campo de la visión o medicina.
- Con el fin de recolectar los datos necesarios para este trabajo, se implementó un sistema de telemetría, que incluye desde los dispositivos de medición, canales de transmisión, bases de datos, aplicaciones de consulta y consumo, resultando en un sistema integrado que seguirá recolectando datos de forma continua para poder elaborar actualizaciones del dataset aquí presentado.
- Como se usaron los equipos Sonoff y Shelly, los cuales su bajo costo, robustez y flexibilidad para enviar transmisiones por el protocolo MQTT permiten la replicabilidad de este sistema.
- El sistema implementado demostró ser lo suficientemente estable y consistente para tener los registros de los dispositivos cada 10 segundos, según lo establecido. Si bien se reportaron algunos retardos y periodos sin datos esto se debe a factores externos de la implementación, como las desconexiones parciales o totales del sistema eléctrico experimentadas en Ecuador durante los meses de noviembre 2023, abril y mayo del 2024.
- Este trabajo presenta un gran esfuerzo por desarrollar datasets orientados a los datos localizados en Ecuador, esto para impulsar el desarrollo de nuevos algoritmos y técnicas que puedan aplicarse en el entorno local, algo que muchas

veces no es factible porque las condiciones de los datasets no pueden ser replicadas o no corresponden con la realidad nacional.

- El dataset CLED, fue presentado y sustentado en una conferencia internacional, lo cual avala la importancia y la rigurosidad técnica con la que se trabajó para su elaboración.
- Gracias al dataset generado y la disponibilidad continua de datos, se pudo probar constantemente y en tiempo real los modelos de pronóstico, facturación y desagregación.
- Se demostró de forma práctica como a partir de las variables de potencia activa se obtiene el consumo energético y se puede usar para distintas aplicaciones, como la facturación del consumo y el pronóstico.
- En este trabajo se demuestran como los esfuerzos open source y de la comunidad de investigadores causan un impacto en futuras investigaciones, en este caso, el proceso de desagregación de cargas toma el proyecto NILMTK y NILMTK-CONTRIB como base para entrenar los modelos con nuestro dataset y obtener las métricas de desempeño de esos modelos. Este proyecto lleva cerca de 10 años en la comunidad y constantemente hay personas aportando para generar nuevos algoritmos, datasets y métricas.
- La revolución en los grandes modelos de lenguaje empezada en 2018 y popularizada en los últimos años gracias a grandes modelos como GPT y LLAMA han dado paso a modelos como CHRONOS, el cual toma la filosofía detrás de los contextos para usarla en la tarea de pronósticos, lo cual es un acercamiento novedoso y presenta resultados aceptables.
- Se diseñó un dashboard para monitoreo en tiempo real y consumo de los datos planteados en este proyecto, esto es un elemento diferenciador frente a otras implementaciones que usan herramientas prediseñadas y que pueden no adecuarse a las necesidades reales de un usuario.

Recomendaciones

- Se pueden encontrar otros dispositivos para transmitir datos sobre otros protocolos como MODBUS, que, si bien pueden ser más caros y presentan una mayor complejidad, presenta una alternativa para la etapa de medición.

- Para este trabajo fue indispensable consultar a profesionales en el área de electricidad, y como probablemente este tipo de perfiles usen el aplicativo desarrollado, antes de llegar a una fase comercial debe ser testeada por estos profesionales para ajustarlo a sus necesidades.
- Se pueden optimizar los tiempos de consulta en las bases de datos si es que se implementan mecanismos como cacheo de datos. Esto es importante en este tipo de aplicaciones ya que los datos históricos no van a cambiar, y los últimos datos pueden ser cacheados para su consumo.
- Para el modelo de pronóstico se puede realizar un preentrenamiento del modelo y escoger otra versión con más parámetros, en este trabajo se usó la versión MINI de 20 Millones de parámetros, sin embargo, están disponibles las versiones SMALL, BASE y LARGE, con 46, 200 y 700 millones de parámetros respectivamente.
- En caso de querer contribuir en un proyecto open source, es importante registrarse en la lista de correos y mantenerse actualizado de los últimos avances y la planificación de desarrollo, esto para contribuir activamente con nuestros propios desarrollos, esto aplica para el proyecto NILMTK donde continuamente se integran nuevas funcionalidades y características.

6. BIBLIOGRAFÍA

- Alsalemi, & Himeur. (2022). Recent trends of smart non-intrusive load monitoring in buildings. *International Journal of Intelligent Systems*.
- Ansari, A. F. (2024). Chronos: Learning the Language of Time Series. *arXiv preprint arXiv:2403.07815*.
- Batra, & Kelly. (2014). NILMTK: an open source toolkit for non-intrusive load monitoring. *e-Energy*.
- HART, G. W. (1992). *Nonintrusive Appliance Load Monitoring*.
- Hastie, T., & Friedman, J. (2001). *The Elements of Statistical Learning*.
- Himeur, & Alsalemi. (2022). Building power consumption datasets :Survey, taxonomy and future directions. *Energy and Buildings*.
- Kelly, J. a. (2014). Metadata for Energy Disaggregation.
- Kelly, J., & Knottenbelt, W. (2015). Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. *BuildSys*.
- Klemenjak, C., & Goldsborough, P. (2016). *Non-Intrusive Load Monitoring: A Review and Outlook*.
- Kolter, Z., & Johnson, M. (2011). REDD: A public data set for energy dissagregation research.
- Kukunuri, & Batra. (2020). NILMTK-Contrib:Towardsreproduciblestate-of-the-artenergydisaggregation.

- Kumar, J., & Gupta, R. (2023). Power consumption forecast model using ensemble learning for smart grid. *The Journal of Supercomputing*.
- Schirmer, P., & Mporas, I. (2023). Non-Intrusive Load Monitoring: A Review. *IEEE TRANSACTIONS ON SMART GRID*.
- Shelly. (2023). Obtenido de <https://www.shelly.com/en/products/shop/shelly-3-em>
- Soni, D. &. (2017). A survey on mqtt: a protocol of internet of things (iot). *International conference on telecommunication, power analysis and computing techniques*.
- SonOff. (2024). Obtenido de <https://sonoff.tech/product/smart-plugs/s31-s31lite/>
- Yan, K., & Wang, X. (2018). Multi-Step Short-Term Power Consumption Forecasting with a Hybrid Deep Learning Strategy. *Energies*.
- Zhang, & Zhong. (2018). Sequence-to-Point Learning With Neural Networks for NILM. *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Zoha, A., & Gluhak, A. (2012). Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey. *Sensors*.