Revista Iberoamericana de Automática e Informática Industrial

Revista Iberoamericana de Automática e Informática Industrial 21 (2024) 243-251

www.revista-riai.org

Estimación del estado de carga de una batería de litio con redes neuronales y validación con FPGA-en-lazo

Erik Martínez-Vera[a]*, Alfredo Rosado-Muñoz[b], Pedro Bañuelos-Sánchez[a]

[a] *Sistemas Inteligentes, Universidad de las Américas Puebla. Ex hacienda Sta. Catarina Mártir S/N. San Andrés Cholula, Puebla. C.P. 72810. México.*
[b] *Dpto. Ingeniería Electrónica. ETSE-UV, Universitat de Valencia. Avda. Universitat s/n 46100 Burjassot. Valencia. España.*

To cite this article: Martínez-Vera, E., Rosado-Muñoz, A., Bañuelos-Sánchez, P. 2024. Lithium-ion Battery State of Charge Estimation with Neural Networks and FPGA-in-the-loop validation. Revista Iberoamericana de Automática e Informática Industrial 21, 243-251. https://doi.org/10.4995/riai.2024.20718

Resumen

Los vehículos eléctricos presentan una alternativa viable para reducir las emisiones de gases tóxicos en las concentraciones urbanas y para disminuir los efectos de los gases de invernadero. La batería de los vehículos eléctricos debe ser monitoreada con precisión para asegurar su funcionamiento adecuado y seguro. Para esto, es necesario desarrollar algoritmos eficientes que permitan estimar de forma precisa el estado de carga mediante dispositivos embarcados en el vehículo. En este trabajo, se utiliza un conjunto de datos de ciclado de una batería de Litio para entrenar una red neuronal para la estimación del estado de carga. Se realiza una optimización bayesiana para establecer la mejor arquitectura de red neuronal y se valida el comportamiento frente a las mediciones reales que ofrece el conjunto de datos. Para su utilización en un dispositivo embarcado, la red neuronal se valida con un modelo de hardware-en-lazo (HIL) en un FPGA con aritmética de punto fijo. Después del entrenamiento se observa un error promedio cuadrático menor al 2% y una precisión promedio del 97.5%.

Palabras clave: Estado de carga, redes neuronales, FPGA, batería de litio, vehículos eléctricos.

Lithium-ion battery state of charge estimation with neural networks and FPGA-in-the-loop validation

Abstract

Electric vehicles present a viable alternative to reduce toxic gas emissions in urban concentrations and to reduce the effects of greenhouse gases. The battery of electric vehicles must be precisely monitored to ensure proper and safe operation. For this, it is necessary to develop efficient algorithms that allow the state of charge to be accurately estimated using devices embedded in the vehicle. In this work, a set of data from cycling a Lithium battery is used to train a neural network for state of charge estimation. A Bayesian optimization is performed to establish the best neural network architecture and the behavior is validated against the actual measurements offered by the data set. For use in an embedded device, the neural network is validated with a hardware-in-loop (HIL) model on an FPGA with fixed-point arithmetic. After training, a root mean square error of less than 2% and an average accuracy of 97.5% are observed.

Keywords: State of charge, neural network, FPGA, lithium-ion battery, electric vehicles.

1. Introducción

Para reducir los efectos del cambio climático y asegurar un futuro sostenible, la organización de las Naciones Unidas ha propuesto diecisiete Objetivos de Desarrollo Sustentable entre los que destacamos los números once y trece: "Ciudades y Comunidades Sostenibles" y "Acción por el clima". Actualmente, las ciudades concentran el 50% de la población mundial, cifra que se estima aumentará al 60% para el 2050. Esto conlleva a que el 70% de las emisiones de carbono se generen en dichos asentamientos (ONU, 2023). La Administración Nacional de Aeronáutica y del Espacio (NASA) reportó al mes de Julio del 2023 como el mes más caluroso desde el inicio de los registros, en 1880 (NASA, 2023). Esto se añade a la cifra récord de incendios forestales

*Autor para correspondencia: erik.martinezva@udlap.mx
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

(CBCNews, 2023; HWMO, 2023; The Guardian, 2023) e inundaciones (EuroNews, 2023) a nivel mundial. Hasta el año 2020, el 99.8% del transporte mundial utilizaba motores de combustión interna como método de propulsión, los cuales generan emisiones de gases de tóxicos y de efecto invernadero (Rosero et al., 2020). El transporte contribuye en alrededor del 25% de las emisiones globales de CO2 (Leach et al., 2020). Los vehículos eléctricos son una propuesta para reducir el consumo de combustibles fósiles y sus emisiones derivadas. Dentro de los componentes del sistema de potencia en un Vehículo Eléctrico (EV por sus siglas en inglés) se encuentra la batería, el motor, los convertidores de potencia y el cableado de alto y bajo voltaje. La Figura 1 muestra un diagrama simplificado del tren de potencia de un EV. De estos componentes, la batería es un elemento que requiere atención especial debido a su alta concentración de energía. Por lo tanto, es importante desarrollar algoritmos que determinen con precisión el estado de ésta. Sin embargo, en una batería, el voltaje, corriente y la temperatura externa son las variables medibles. Entonces, es a partir de ellas que se deben calcular los parámetros que indiquen la cantidad de energía remanente, así como el estado de salud de la misma.

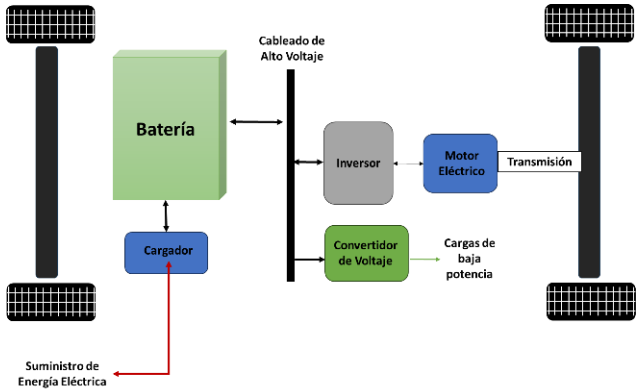

Figura 1: Diagrama de bloques simplificado del sistema de potencia de un vehículo eléctrico (EV).

El Sistema de Administración de la Batería (BMS) regula el funcionamiento adecuado y seguro durante la carga y descarga. Uno de los parámetros críticos que monitorea es el Estado de Carga (SOC) para indicar el nivel de carga restante. Existen diferentes métodos para estimar el SOC los cuales pueden clasificarse en métodos por "Conteo de Coulombs", por "Modelo Equivalente" o "Basados en Datos" (Liu et al., 2022). El estado de carga puede calcularse por el método de Conteo de Coulomb de acuerdo con la Ecuación (1),

$$z = z_0 - \frac{1}{C_N}\int I_L dt \qquad (1)$$

donde z es el estado de carga, z_0 es el estado de carga inicial, C_N es la capacidad nominal de la batería e I_L es la corriente de la carga. Este método es preciso en condiciones controladas de laboratorio, pero su implementación es impráctica en un EV ya que es necesario descargar completamente la batería para iniciar el conteo si se desconoce el estado de carga inicial.

Los métodos por Modelo Equivalente utilizan un circuito Equivalente o espectroscopía de impedancia electroquímica para caracterizar la batería e incluir el estado de carga como una de las variables (Martinez-Vera et al., 2019). Dentro de esta categoría también se incluyen los métodos que utilizan filtros para estimar el valor futuro de SOC a partir del modelo y mediciones anteriores. Uno de los más comunes es el filtro de Kalman (Cui, Hu, et al., 2022; Shenghao, 2021). Sin embargo, en ambos casos la estimación será tan precisa como el modelo de la batería. Los modelos más precisos requieren modelos matemáticos complejos con demasiados parámetros y difíciles de generalizar.

Los métodos basados en datos están basados en identificar las relaciones inherentes en las mediciones de los parámetros de entrada y salida durante el funcionamiento de la batería. Entonces, se aplican algoritmos de aprendizaje de máquina, como las redes neuronales o máquinas de soporte vectorial, para aprender dichas relaciones y reproducirlas de forma independiente. En (Gao et al., 2017) se emplea una red neuronal auto recurrente con wavelets como función de activación en los nodos de la capa oculta. En (Kang et al., 2014) una red neuronal con función de base radial se utiliza para estimar el SOC incluyendo el proceso de degradación de la batería. Las redes neuronales también se han combinado con los filtros Kalman para la estimación del SOC (Cui, Dai, et al., 2022). Una red neuronal de largo-corto término es comparada contra dos redes de feed-forward (Almaita et al., 2022) y optimizada con algoritmos Bayesianos en (Yang et al., 2022).

En un sistema dinámico, como es el caso de un EV, el SOC debe monitorearse con precisión para evitar situaciones de sobrevoltaje, bajo voltaje, sobrecarga o carga reducida que puedan dañar la batería. Entonces, los algoritmos de monitoreo deben ser validados en tiempo real. Sin embargo, los ciclos de carga y descarga de una batería de alto voltaje representan un riesgo para los usuarios. Además, dichos ciclos alteran los parámetros físicos de las baterías por lo que afectan la reproducibilidad de los análisis.

Las técnicas de Hardware-In-the-Loop (HIL) permiten realizar análisis de algoritmos de control en condiciones controladas y reproducibles. El procedimiento consiste en reemplazar un componente físico por un modelo que es simulado en lazo cerrado con un componente de hardware como si se tratase del microprocesador de la Unidad Electrónica de Control (ECU) de un vehículo. De esta forma, es posible validar el algoritmo de control sin necesidad de arriesgar a los operadores a condiciones de alto voltaje y corriente. Además, es posible identificar posibles fallos en el código de forma anticipada a las pruebas con sistemas reales acelerando el proceso de diseño y validación. Los dispositivos FPGA también pueden ser utilizados en simulaciones de lazo cerrado para la validación de algoritmos (Ząd ek et al., 2015). En (Ali et al., 2022) se utilizó un FPGA para la implementación del sistema de estimación del estado de carga de la batería de un vehículo eléctrico.

En este trabajo, se describen los algoritmos desarrollados para la estimación del estado de carga de una batería de litio empleando un método "Basado en Datos" con redes neuronales. El resto de este artículo se divide de la siguiente forma: en la sección 2 se presentan los datos para entrenar y validar el modelo. Se emplea un conjunto de datos estándar de

uso generalizado como comparativa y similitud con el comportamiento de las baterías de un EV. También, Se analiza el comportamiento de la red neuronal y se compara con otros algoritmos de estimación del estado de carga de batería (SOC). En la sección 3, para validar el correcto funcionamiento en un sistema embarcado como el caso de un EV, se realiza la implementación en un dispositivo hardware de tipo FPGA y se valida el modelo de computación en punto fijo mediante la evaluación del modelo a través de HIL. Además, se presenta la comparativa ante otros métodos de estimación del estado de carga. Por último, la sección 4 concluye este trabajo incluyendo las propuestas para futuras implementaciones.

2. Diseño de la Red Neuronal

2.1 Datos de Entrenamiento

Los datos utilizados para el entrenamiento de las redes neuronales en este trabajo corresponden a la corriente, voltaje, estado de carga (SOC) y temperatura de una batería de Li[NiMnCo]O2 con voltaje nominal de 3.6 V, capacidad nominal de 3.0 Ah y densidad de energía de 240 Wh/kg. Este conjunto de datos es de libre acceso, disponible en (Kollmeyer et al., 2020; Vidal et al., 2019). El conjunto de datos está creado aplicando pruebas estandarizadas de carga y descarga como la de Caracterización de Pulso y Potencia Híbrido (HPPC) (Christophersen, 2015). También, realizan múltiples pruebas de descarga con diferentes perfiles de manejo como el de Dinamómetro Urbano (UDDS), el de Economía en Autopista (HWFET), el LA92 y US06 variando la temperatura ambiente desde -20ºC hasta 40ºC (Liu et al., 2022; US EPA, 2023).

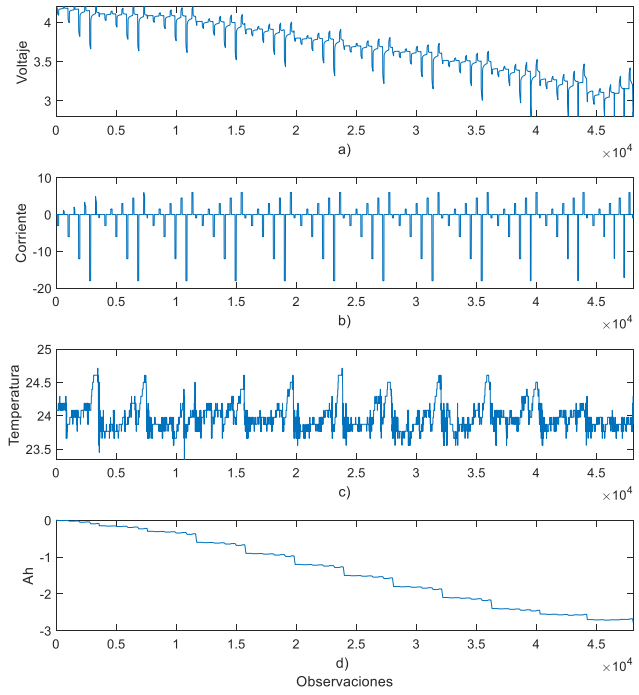

Figura 2: Perfil HPPC de carga y descarga de batería de litio a 25ºC: a) voltaje, b) corriente, c) Temperatura de la batería y d) Ah (Kollmeyer et al., 2020).

Estos perfiles reproducen las condiciones a las que un vehículo está sometido durante condiciones reales de manejo. Para cada prueba, se reportan los valores en el tiempo: corriente, voltaje, Ah y temperatura de la batería.

La Figura 2 muestra el perfil HPPC de voltaje, corriente, temperatura y Ah de la batería a temperatura ambiente de 25ºC. De acuerdo con la Figura, la corriente oscila entre valores positivos y negativos indicando carga y descarga de la batería. En la sección inferior de la Figura se observa como el valor de Ah disminuye hasta utilizar la capacidad total de la batería.

La Figura 3 muestra la descarga de la batería con el perfil de manejo US06 a temperatura ambiente de 10ºC. Como se mencionó, este perfil simula las condiciones de manejo a las que está sometido un vehículo. Puede observarse que el comportamiento de la corriente es aleatorio, a diferencia de la Figura 2, donde la corriente se comporta con patrones repetitivos.

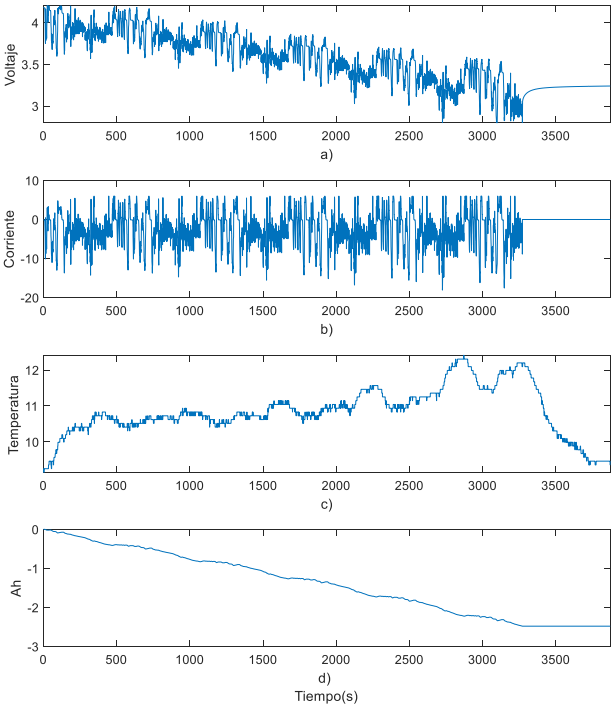

Figura 3: Descarga de la batería de litio con el perfil de manejo US06 a 10ºC: a) voltaje, b) corriente, c) Temperatura de la batería y d) Ah (Kollmeyer et al., 2020).

La Figura 4 muestra los datos de entrenamiento. Puede observarse la concatenación y normalización de múltiples secuencias de carga y descarga a diferentes valores de temperatura. Múltiples pruebas con diferentes perfiles de manejo a temperaturas de -10ºC, 0ºC, 10ºC y 25ºC fueron ensamblados en un solo conjunto de datos. Posteriormente, se calcula el estado de carga dividiendo los datos Ah entre la capacidad nominal de la batería. El término Ah se refiere al flujo de corriente eléctrica con magnitud de un Amperio por una hora. Dado que un Amperio equivale un Coulomb por segundo, un Amperio-hora equivale a 3600 Coulombs, es decir, el término Ah indica magnitud de carga eléctrica. De acuerdo con la Ecuación (1), la integral de I_L en el tiempo es equivalente a calcular la cantidad de carga.

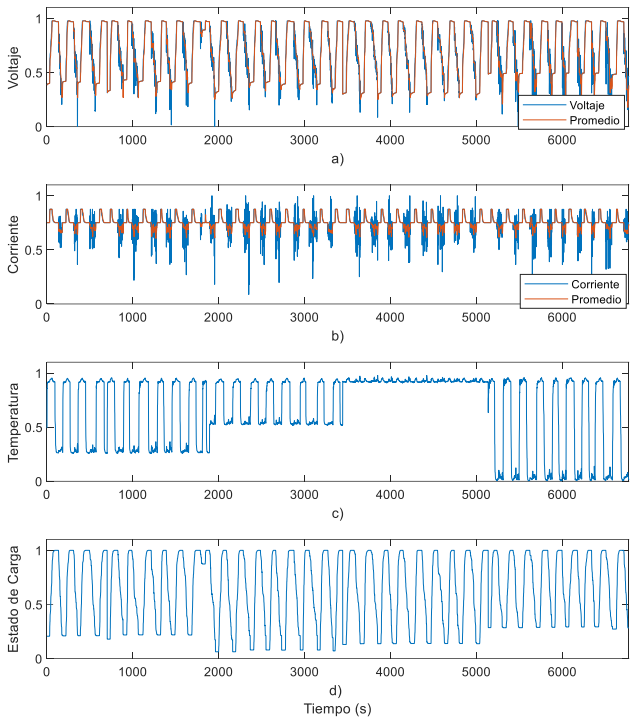

Figura 4: Datos para entrenamiento a) Voltaje, b) corriente, c) temperatura y d) estado de carga (Kollmeyer et al., 2020).

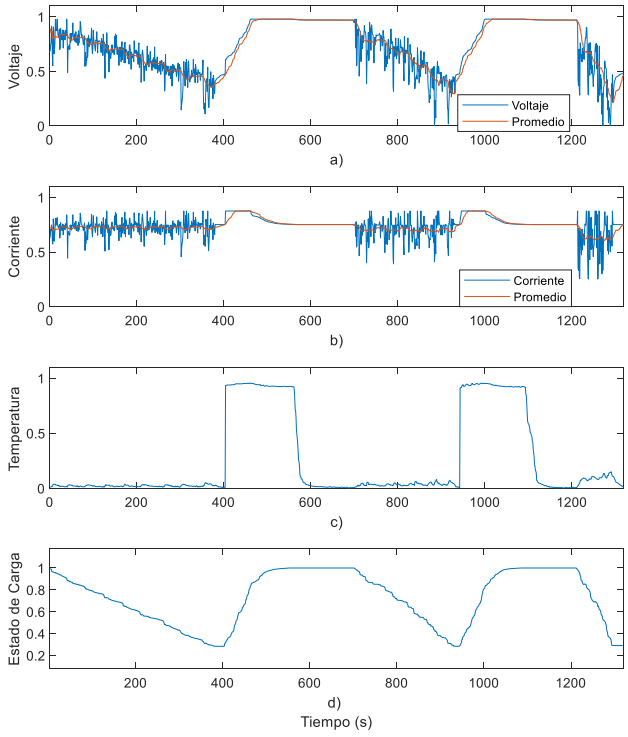

Figura 5: Datos para validación a) Voltaje, b) corriente, c) temperatura y d) estado de carga (Kollmeyer et al., 2020).

Al dividir entre la capacidad nominal de la batería, C_N, se obtiene el valor del estado de carga (SOC) el cual se reporta normalmente normalizado entre [0,1] equivalente a valores entre 0 y 100% de la carga de la batería. En las Figuras 2d y 3d, se observa el comportamiento de la carga de la batería con unidades de Ah. A partir de la Figura 4d, se observa que la magnitud de carga ya se ha procesado dividiendo el valor de Ah entre la capacidad nominal de la batería por lo que ahora el rango varía entre [0,1].

El resto de los datos se normalizan y se dividen en segmentos de entrenamiento, pruebas y validación (Kollmeyer et al., 2020). De acuerdo al procedimiento en (MathWorksFPGA, n.d.), un total de seis variables con 8092 observaciones cada una se utilizan para el entrenamiento y validación. Este conjunto de datos fue dividido en el grupo de entrenamiento con 84% de las observaciones y el grupo de validación con el 16% restante. Las cinco primeras columnas corresponden al voltaje, corriente, promedio del voltaje, promedio de la corriente y temperatura. Estas corresponden a los datos de entrada denominados como predictores. La última columna corresponde al estado de carga de la batería el cual se emplea como dato a validar o etiqueta real del SOC. La Figura 5 muestra los datos de validación donde también pueden observarse la concatenación y normalización de múltiples secuencias de carga y descarga a diferentes valores de temperatura.

2.2 Entrenamiento y Optimización de la Red Neuronal

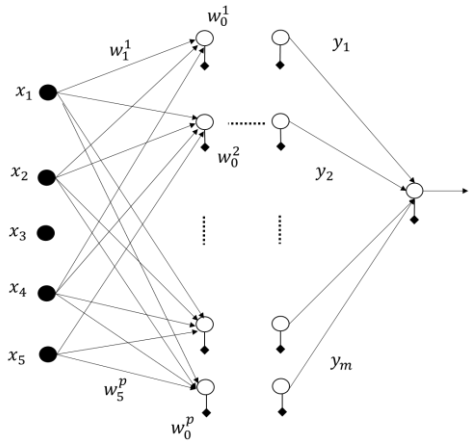

Figura 6: Arquitectura de la red Neuronal con cinco variables de entrada, y N capas ocultas.

La arquitectura de la red neuronal implementada se muestra en la Figura 6 y corresponde a una red tipo Feed Forward con cinco variables de entrada, una variable de salida y tres capas ocultas con 53, 2 y 6 neuronas, respectivamente. La relación entre los datos a la entrada, las capas ocultas y la salida, está determinada de acuerdo a las Ecuaciones (2) y (3),

$$y = h\left(\sum_{m=1}^{M} w_m H_m + b_{2m}\right) \tag{2}$$

$$H_j = h\left(\sum_{m=1}^{M} w_{mp} x_m + b_{1p}\right), \qquad \forall p = \{1, \dots, P\} \tag{3}$$

donde y es la salida de la red neuronal, h es la función de activación, H_j es la salida de las capas ocultas, w_{mp} corresponde a los pesos y b_{1p} a los valores de la ordenada. El voltaje, corriente, temperatura, voltaje promedio y corriente promedio son los datos de entrada que corresponden x_m en la Ecuación (3), con el estado de carga como la salida de la red, correspondiente a y en la Ecuación (2).

El algoritmo de entrenamiento empleado es el método Broyden-Fletcher-Goldfarb-Shanno de memoria limitada (L-BFGS) para minimizar el Error de Promedios Cuadráticos (MSE) (MathWorksNN, 2023). Este algoritmo es un método numérico Quasi-Newtoniano utilizado para resolver problemas de optimización no lineales. Su ventaja consiste en que utiliza una estimación de la matriz Hessiana evitando así su cálculo y el de su inversa. (Nocedal & Wright, 2006; Su, 2023). Otros parámetros que influyen durante el entrenamiento de la red neuronal incluyen la función de activación, el número de capas ocultas, la cantidad de neuronas en cada capa, el parámetro de ajuste de la regularización L2, y lambda. Este último es un coeficiente agregado a la función de pérdida durante el entrenamiento de la red. Un valor de lambda muy alto ocasiona una falta de ajuste (under-fitting) durante el entrenamiento y disminuye la capacidad de generalización de los resultados durante la validación. Para la salida final de cada neurona el empleo de diferente función de activación modifica el comportamiento. Por ello, las funciones de activación evaluadas son ReLU, tanh y Sigmoid, definidas en las Ecuaciones (4), (5) y (6),

$$\text{ReLU} = \begin{cases} 0, & if\ x \leq 0 \\ x, & if\ x > 0 \end{cases} \quad (4)$$

$$\text{Sigmoid} = \frac{1}{1 + e^{-x}} \quad (5)$$

$$\tanh = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (6)$$

2.3. Optimización Bayesiana

Para determinar el resto de los parámetros, se utilizó la optimización Bayesiana. Este tipo de optimización es útil cuando se tiene un espacio de búsqueda demasiado vasto para navegarlo manualmente. En nuestro caso, se busca encontrar el número de capas ocultas, el número de neuronas por capa, la función de activación y el valor del parámetro de regulación que minimicen el valor del MSE. Este procedimiento consiste en localizar un punto que minimice una función objetivo escalar dentro y u un dominio limitado. Para esto, se define un modelo probabilístico que captura el conocimiento del comportamiento de la función objetivo y un modelo observacional que describe el método de generación de los datos. También, se define una función de perdida que describe que tan optima es la secuencia generada. Para cada estimación de esta función, se actualiza el precedente del modelo probabilístico, lo que resulta en una mejor distribución del posterior (Frazier, 2018; Shahriari et al., 2016; Snoek et al., 2012).

La Tabla 1 muestra un extracto de los resultados de 5 iteraciones (estimaciones) donde cada combinación de variables minimiza la función objetivo (Ecuación 7). Al finalizar las iteraciones, la combinación con el menor valor de la función objetivo es seleccionada como la solución óptima.

Tabla 1: Ejemplos de parámetros de entrenamiento. La red neuronal elegida para implementación está marcada con **.

Función de Activación	Estandarización	Lambda	Capas, Número y dimensión
Relu	Sí	2.93e-8	2, [1 50]
Sigmoid	Sí	0.0028	2, [256 167]
Relu**	Sí**	3.7e-5**	3, [53 2 6]**
Tanh	No	1.7e-9	1, [1]
Ninguna	No	3.82e-9	2, [3 8]

$$\log(1 + cross - validation\ loss) \quad (7)$$

Para la validación cruzada se dividen los datos de entrenamiento en 5 segmentos, los cuales son intercambiados durante entrenamiento y validación. La Figura 7 muestra la evaluación del mejor valor observado de la función objetivo para cada iteración. En la primera iteración se observa que el valor de la función objetivo es de $1.6x10^{-3}$. Sin embargo, en la quinta iteración se observa un valor aún menor de la función objetivo. Entonces, el primer resultado corresponde a una solución local al problema de optimización. Es decir, dentro de el universo de búsqueda, es posible encontrar una solución mejor la cual se encuentra fuera de la vecindad de la solución actual. El número total de iteraciones fue de 30 y se definió de forma empírica al observar más de 20 iteraciones sin encontrar una mejor solución.

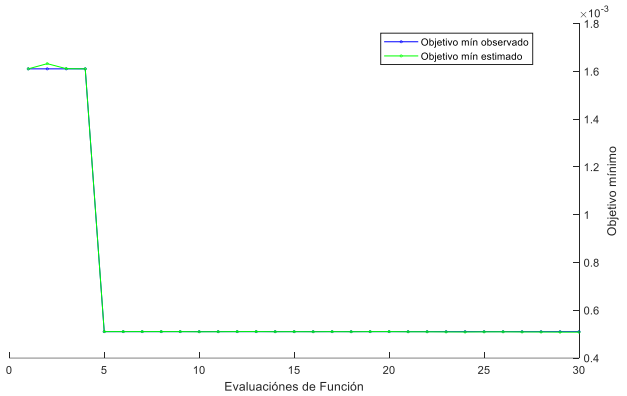

Figura 7: Evolución de optimización Bayesiana para determinar los parámetros óptimos de la red neuronal.

2.4. Validación de la Red Neuronal

De acuerdo a la optimización, los parámetros marcados con ** en la Tabla 1 entregan el menor valor de la función objetivo. Estos fueron seleccionados para entrenar la red neuronal. Una vez que se han determinado los parámetros óptimos de entrenamiento, es posible validar el funcionamiento de la red con el segmento de datos reservado para este propósito. La métrica empleada para validar el desempeño de la red neuronal

es el error cuadrático promedio menos 1 donde el MSE se define en la Ecuación (8),

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(X_i - Y_i)^2 \qquad (8)$$

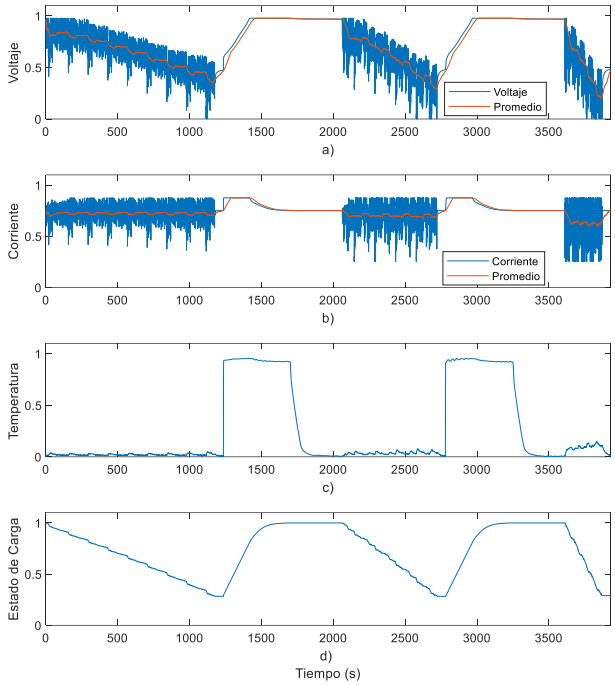

Figura 8: Datos para validación en Simulink, a) voltaje, b) corriente, c) temperatura y d) estado de carga de acuerdo a los datos reportados en [Kollmeyer2020].

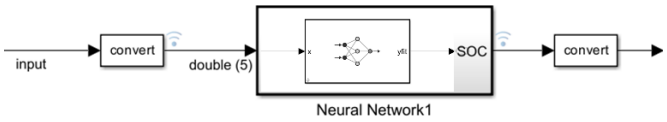

Figura 9: Diagrama de bloques en Simulink para la red neuronal de estimación del estado de carga de la batería.

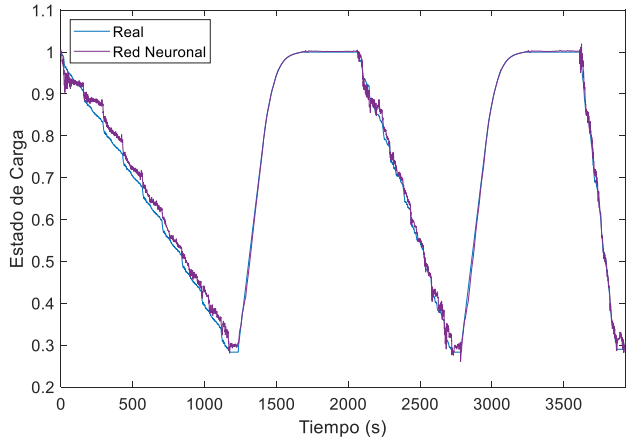

Figura 10: Estimación del estado de carga en Simulink.

Para la configuración optimizada, este valor es igual 0.9997. Una vez que se ha verificado el correcto funcionamiento de la red neuronal, se incorpora un nuevo conjunto de datos para validación en Simulink. Estos datos se muestran en la Figura 8. Puede observarse que este conjunto consta de más de 30,000 puntos con distintos ciclos de carga y descarga de la batería.

En la Figura 9 puede observarse la implementación en Simulink de la red neuronal. Los resultados de la simulación se encuentran en la Figura 10. Puede observarse que la estimación del estado de carga sigue la tendencia general de la medición. La máxima diferencia absoluta porcentual obtenida en este caso es igual a 0.1603.

3. Simulación FPGA-en-lazo

Antes de exportar el modelo de la red neuronal al FPGA, es necesario modificar los parámetros de operación de la simulación. El primero es cambiar el tipo de datos de punto flotante (doble precisión) a punto fijo. Este paso permite reducir la carga computacional del hardware, si bien puede introducir errores adicionales. En este caso, las señales de entrada y salida se transformaron a señales sin signo con 16 bits de longitud de palabra, 16 y 15 bits de longitud para la parte entera y fraccionaria, respectivamente. El procedimiento para cambiar el tipo de datos consiste primero en simular con datos de doble precisión para determinar posibles rangos de valores. Después, se propone una longitud de palabra adecuada de acuerdo a los valores observados y se vuelve a simular. Sí el resultado de la simulación concuerda con los resultados esperados, se acepta el cambio de datos, de otra forma, se vuelven a proponer nuevos tipos de datos. La Figura 11 muestra el paso inicial de este proceso donde la columna "Name" indica cada uno de los bloques dentro de la red neuronal, la columna "SpecifiedDT" indica el valor actual del tipo de datos para cada bloque y la columna "ProposedDT" indica el valor propuesto de tipo datos.

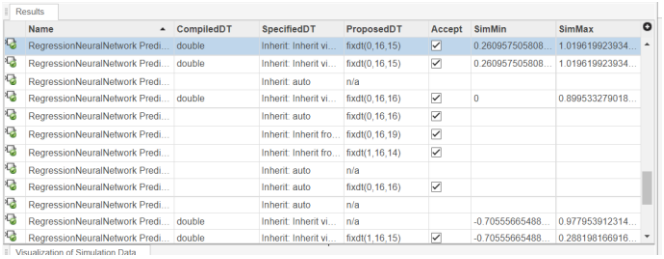

Figura 11: Optimización para cambio de tipo de datos de punto flotante a punto fijo.

La Figura 12 muestra la estimación del estado de carga con los resultados en punto fijo. Se observa que se produce una ligera diferencia entre la simulación con diferentes tipos de datos, manteniendo eso sí la tendencia general. La máxima diferencia absoluta porcentual entre ambas simulaciones es del 2.2%. Esto se traduce en una diferencia máxima de 0.1643 entre la medición real dada por el conjunto de datos, y la ejecución del algoritmo empleando punto fijo.

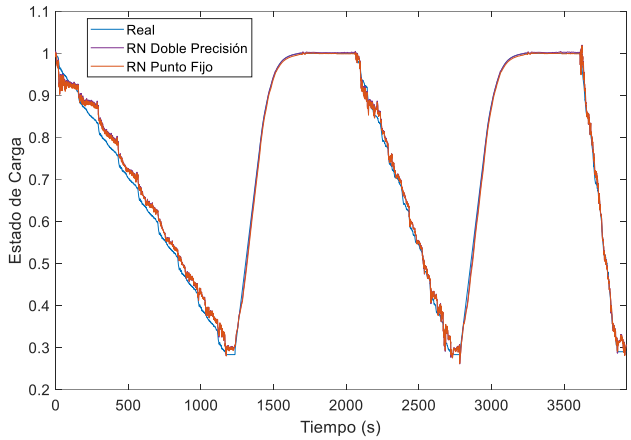

Figura 12: Comparación de la estimación del estado de carga (SOC) entre la medición, la estimación con datos tipo flotante y tipo punto fijo.

Una vez que se ha validado el cambio de datos, el modelo de la red neuronal puede exportarse al FPGA. Para esto, es necesario generar el código VHDL, en este caso se emplea la herramienta de síntesis Vivado 2022.1 y una tarjeta Zedboard que incluye un FPGA Xilinx XC7Z020-1CLG848C Zynq-7000 AP SoC como hardware para implementación. La Figura 13 muestra el resumen de la generación del código VHDL. En la columna izquierda se observa la generación de un archivo de código fuente para cada componente dentro de la red neuronal, por ejemplo, capas ocultas 1, 2 y 3. En la parte inferior se observa un diagrama a bloques de cada componente de la red. En este caso se muestran los bloques de la capa oculta número 1: weights, bias y activation. En la parte superior es posible observar el código generado para el componente seleccionado, en este caso resaltado en azul. El código es generado con la herramienta "HDL Workflow Advisor" de Matlab. De esta forma, es posible verificar que el código será compatible tanto con el hardware deseado y la simulación FIL.

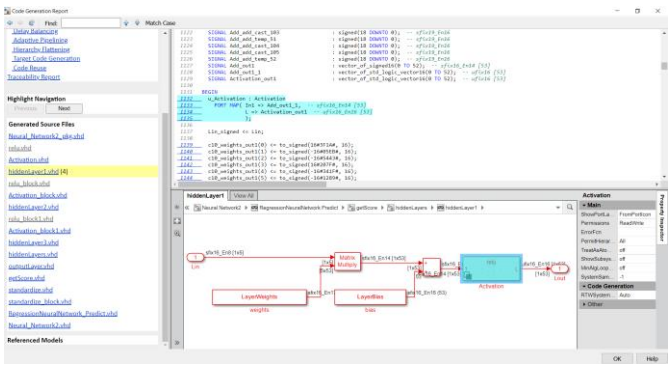

Figura 13: Detalle de la generación del código VHDL para la red neuronal

Para disponer de un sistema HIL completo y de lazo cerrado, es necesario crear una interfaz entre la simulación en la computadora y el FPGA. Es preciso indicar la velocidad del reloj en el FPGA, así como las características de las señales de entrada y salida. En este caso se seleccionó la velocidad del reloj en 100 MHz. Para la señal de salida, se utiliza un tipo de datos de señal sin signo, con longitud de palabra de 16 bits y longitud fraccionaria de 15 bits.

Con esta información, la herramienta de síntesis genera el archivo de flujo de bits necesario para programar el FPGA. La Figura 14 muestra el FPGA-en-lazo con la computadora y los resultados que genera el hardware visualizado en la pantalla.

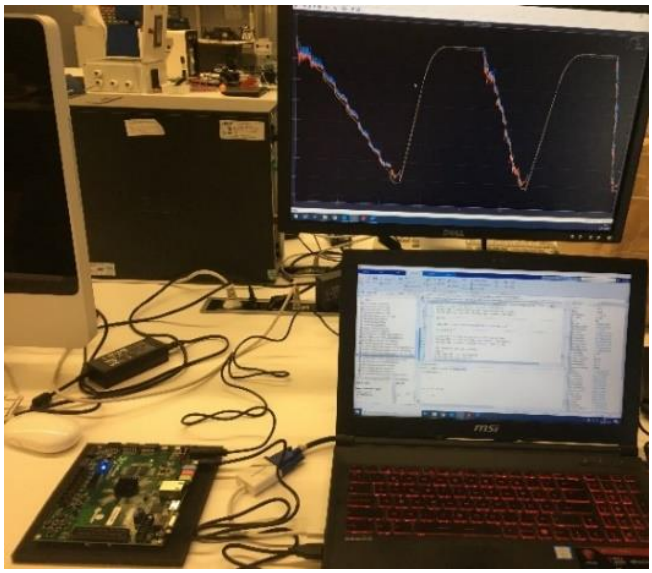

Figura 14: FPGA-en-lazo con la computadora para estimación del estado de carga.

La Figura 15 muestra la utilización de recursos una vez que se ha generado el proyecto. Como se puede observar, la utilización de recursos es baja excepto en las unidades de cómputo de procesado digital de señal (DSP) ya que de éstas depende la computación de las neuronas, basada principalmente en multiplicación y acumulación.

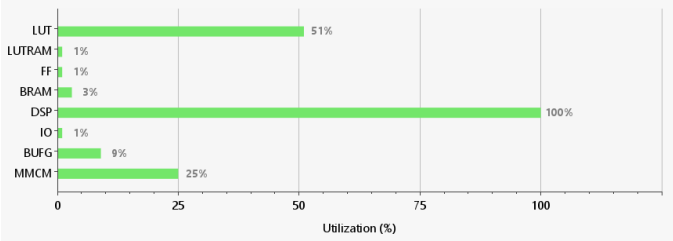

Figura 15: Gráfica de utilización de recursos en el FPGA Xilinx XC7Z020-1CLG848C Zynq-7000 AP SoC:

La Figura 16 muestra los resultados de la simulación con FGPA-en-lazo. Se incluye la medición del estado de carga y la simulación con punto fijo sin FPGA. Puede observarse que las curvas del FPGA y punto fijo son casi idénticas. La máxima diferencia absoluta entre la simulación con FPGA y las mediciones reales del conjunto de datos es de 0.1825. Se observa una ligera disminución en la precisión de la simulación al transferir de tipo de datos de doble precisión a punto fijo en FPGA. Esto es un claro indicador de la problemática que se puede presentar al migrar hacia un dispositivo embarcado debido a la reducción de recursos disponibles para la implementación del algoritmo. Entonces, se demuestra la

importancia de la validación en hardware de los algoritmos de estimación ya que se asegura su funcionalidad práctica más allá de la simulación en la computadora.

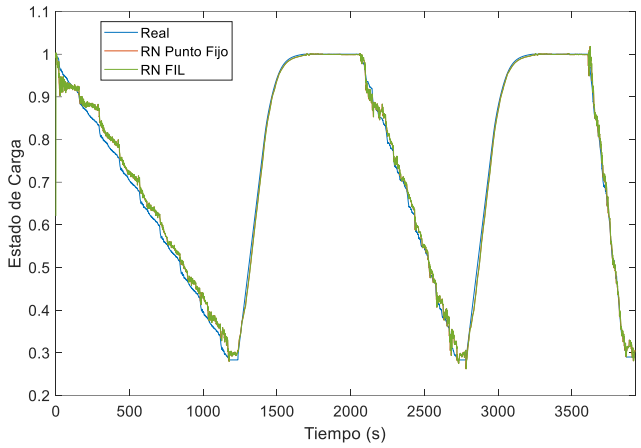

Figura 16: Comparación de la Estimación del estado de Carga entre la medición, la estimación con datos de punto fijo y con FPGA-en-lazo (HIL).

Como comparativa a los resultados obtenidos en este trabajo, la Tabla 2 presenta un resumen de métodos recientes de estimación del estado de carga indicando los perfiles de manejo, tipo de batería y métrica de error reportada. Los resultados muestran que este trabajo obtiene unos valores de error similares a otros trabajos, si bien ligeramente superiores.

También en la Tabla 2 se muestra la comparativa de las arquitecturas de red de los trabajos relacionados. Puede observarse que los algoritmos que obtienen los menores RMSE presentan arquitecturas más complejas con una o más capas ocultas LSTM o recurrentes además de capas de conexión completa de entrada y salida.

Durante el traslado del algoritmo del ordenador hacia el FPGA, fue necesario transformar el tipo de datos de doble precisión a punto fijo. Se observó que el algoritmo podría representarse con longitud de palabra de 22 bits, mayor al valor seleccionado de 16 bits. Aunque se obtendría mayor precisión en la representación de los valores de simulación, la generación del código VHDL requería 20,000 celdas lógicas adicionales a las disponibles en el FPGA destino. De esta forma, se comprueba que el proceso de migración desde la simulación en ordenador hacia el hardware embarcado, esta limitado por la capacidad de los recursos disponibles. Adicionalmente debe señalarse que el algoritmo esta destinado a un vehículo eléctrico. Entonces, un aumento en la capacidad del hardware se reflejará en un aumento del precio de los ya muy costos vehículos eléctricos. Por lo tanto, debe considerarse sí la disminución del RMSE del valor obtenido en este trabajo (1.733%) al mejor valor reportado (0.89%) justifica el aumento de recursos necesario. Aunque es conocido que las redes LSTM son de naturaleza más compleja que las más sencillas redes feed forward, mayor análisis se necesita para validar si estas propuestas pueden ser implementadas en hardware manteniendo los valores de RMSE reportados sin aumentar los requerimientos del hardware

Tabla 2: Comparación entre métodos existentes y el método propuesto

Ref.	Algoritmo	RMSE (%)	Perfil de Manejo	Batería	Arquitectura de Red	Entrenamiento
(Yang et al., 2022)	BiLSTM	0.89	FUDS, US06	INR18650-20R 2 Ah, 3.6V	Dos capas de entrada y salida en conexión completa + dos capas ocultas LSTM de 16 hasta 256 neuronas	Optimización Adam y Bayes
(Vidal et al., 2019)	DNN-TL	1	UDDS, LA92, US06	LG HG2 3Ah	2 capas ocultas, 1 LSTM de 10 neuronas y otra de conexión completa de 1 neurona	Red Neuronal recurrente con transferencia de aprendizaje
(Hun et al., 2022)	RNN-SA	1.31	UDDS, LA92, US06, HWFET	LiPO 5Ah, 3.7V	Dos capas ocultas en conexión completa y un bloque de RNN con autoatención – (3 capas)	Optimización Adam
(Tiwari et al., 2022)	FF-LM	1.9902	UDDS, LA92, US06	LG HG2 3Ah	Feed forward con dos capas ocultas de 20 neuronas	Levenberg-Marquardt
Este trabajo	FF-OB-FIL	1.7339	UDDS, LA92, US06	LG HG2 3Ah	Feed Forward con 3 capas ocultas de 53, 2, 6 neuronas	Optimización Bayesiana con L-BFGS

4. Conclusiones

En este trabajo se entrenó una red neuronal con cinco entradas y tres capas ocultas para estimar el estado de carga de una batería de Litio. El número de capas y de neuronas en cada capa fue determinado, junto con la función de activación y el parámetro de regularización L2 a través de un algoritmo de optimización Bayesiana. Para el entrenamiento de la red se utilizaron datos de un batería de Litio los cuales se dividieron en entrenamiento y validación. Tras la conversión de los datos de la simulación de punto flotante a punto fijo, se generó el código VHDL para la implementación hardware en FPGA y la interfaz entre la computadora. Con esto, fue posible validar con simulación FPGA-en-lazo el algoritmo de red neuronal para estimación del estado de carga. Este trabajo muestra cómo es posible el empleo de una red neuronal para estimar el estado de carga en dispositivos hardware que permitan hacerlo en sistemas embarcados pues se ha validado la conversión a punto fijo, sin que ello suponga penalización en el desempeño.Si bien los resultados obtenidos están en línea con otros autores, en el

futuro se estudiará la posibilidad de implementar redes neuronales que permitan disminuir la diferencia máxima absoluta entre las estimaciones y las mediciones del estado de carga.

Agradecimientos

Este trabajo ha sido realizado gracias al apoyo de la Universitat de València, España, la Universidad de las Américas-Puebla (UDLAP) y el Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCyT), México.

Referencias

Ali, A., Faisal, N., Zia, Z., Makda, I., & Usman, A. (2022). Rapid Prototyping of Bidirectional DC-DC Converter Control using FPGA for Electric Vehicle Charging Applications. 2022 IEEE 13th International Symposium on Power Electronics for Distributed Generation Systems (PEDG), 1–6. https://doi.org/10.1109/PEDG54999.2022.9923288

Almaita, E., Alshkoor, S., Abdelsalam, E., & Almomani, F. (2022). State of charge estimation for a group of lithium-ion batteries using long short-term memory neural network. Journal of Energy Storage, 52, 104761. https://doi.org/10.1016/J.EST.2022.104761

CBCNews. (2023). Five charts to help understand Canada's record-breaking wildfire season | CBC News. https://www.cbc.ca/news/climate/wildfire-season-2023-wrap-1.6999005

Christophersen, J. P. (2015). Battery Test Manual For Electric Vehicles, Revision 3. https://doi.org/10.2172/1186745

Cui, Z., Dai, J., Sun, J., Li, D., Wang, L., & Wang, K. (2022). Hybrid Methods Using Neural Network and Kalman Filter for the State of Charge Estimation of Lithium-Ion Battery. Mathematical Problems in Engineering, 2022. https://doi.org/10.1155/2022/9616124

Cui, Z., Hu, W., Zhang, G., Zhang, Z., & Chen, Z. (2022). An extended Kalman filter based SOC estimation method for Li-ion battery. Energy Reports, 8, 81–87. https://doi.org/10.1016/J.EGYR.2022.02.116

EuroNews. (2023). Libya, Greece, Brazil: Climate-driven storms cause catastrophic flooding around the world | Euronews. https://www.euronews.com/green/2023/09/13/libya-greece-brazil-climate-driven-storms-cause-catastrophic-flooding-around-the-world

Gao, A. Y., Zhang, F. L., Fu, Z. M., Zhang, Z. C., & Li, H. Di. (2017). The SOC estimation and simulation of power battery based on self-recurrent wavelet neural network. Proceedings - 2017 Chinese Automation Congress, CAC 2017, 2017-January, 4247–4252. https://doi.org/10.1109/CAC.2017.8243525

Hun, R., Xu, J., Sun, M., Zhang, S., Chen, Y., & Chiang, P. Y. (2022). Hardware-Software Co-design of Efficient Light-weight Self-Attention Neural Network for Lithium-Ion Batteries State-of-Charge Estimation. 11th International Conference on Communications, Circuits and Systems, ICCCAS 2022. https://doi.org/10.1109/ICCCAS55266.2022.9825471

HWMO. (2023). Wildfire in Hawaii Factsheet — Hawaii Wildfire Management Organization. https://www.hawaiiwildfire.org/fire-resource-library-blog/wildfire-in-hawaii-factsheet

Kang, L. W., Zhao, X., & Ma, J. (2014). A new neural network model for the state-of-charge estimation in the battery degradation process. Applied Energy, 121, 20–27. https://doi.org/10.1016/J.APENERGY.2014.01.066

Kollmeyer, P., Vidal, C., Naguib, M., & Skells, M. (2020). LG 18650HG2 Li-ion Battery Data and Example Deep Neural Network xEV SOC Estimator Script. 3. https://doi.org/10.17632/CP3473X7XV.3

Leach, F., Kalghatgi, G., Stone, R., & Miles, P. (2020). The scope for improving the efficiency and environmental impact of internal combustion engines. Transportation Engineering, 1, 100005. https://doi.org/10.1016/J.TRENG.2020.100005

Liu, Y., He, Y., Bian, H., Guo, W., & Zhang, X. (2022). A review of lithium-ion battery state of charge estimation based on deep learning: Directions for improvement and future trends. Journal of Energy Storage, 52, 104664. https://doi.org/10.1016/J.EST.2022.104664

Martinez-Vera, E., Marco, J., Ramirez-Cortes, J. M., & Rangel-Magdaleno, J. (2019). Development of a Lithium-ion Battery Model and State of Charge Estimation Algorithm with Hardware-in-the-loop Validation. 2019 IEEE International Instrumentation & Measurement Technology Conference (I2MTC), 57–61. https://doi.org/10.1109/I2MTC.2019.8827050

MathWorksFPGA. (n.d.). Deploy Neural Network Regression Model to FPGA/ASIC Platform - MATLAB & Simulink - MathWorks España. Retrieved October 26, 2023, from https://es.mathworks.com/help/stats/deploy-neural-network-regression-model-to-fpga-platform.html

MathWorksNN. (2023). Train neural network regression model - MATLAB fitrnet - MathWorks España. https://es.mathworks.com/help/stats/fitrnet.html#mw_7e3de6cb-15e5-434e-82ed-8d870213d539_sep_shared-HyperparameterOptimizationOptions

NASA. (2023). NASA Clocks July 2023 as Hottest Month on Record Ever Since 1880 - NASA. https://www.nasa.gov/news-release/nasa-clocks-july-2023-as-hottest-month-on-record-ever-since-1880/

Nocedal, J., & Wright, S. J. (2006). Numerical Optimization. Springer New York. https://doi.org/10.1007/978-0-387-40065-5

ONU. (2023). Ciudades - Desarrollo Sostenible. https://www.un.org/sustainabledevelopment/es/cities/

Rosero, F., Fonseca, N., López, J. M., & Casanova, J. (2020). Real-world fuel efficiency and emissions from an urban diesel bus engine under transient operating conditions. Applied Energy, 261, 114442. https://doi.org/10.1016/J.APENERGY.2019.114442

Shenghao, W. (2021). Improved Algorithm Combining Wavelet Transform and Kalman Filter for Estimating SOC of Lithium-ion Battery in Vehicle. 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering, ICBAIE 2021, 175–179. https://doi.org/10.1109/ICBAIE52039.2021.9390009

Su, J. (2023). Quasi-Newton methods - Cornell University Computational Optimization Open Textbook - Optimization. https://optimization.cbe.cornell.edu/index.php?title=Quasi-Newton_methods

TheGuardian. (2023). A visual guide to Greece's deadly wildfires | Wildfires | The Guardian. https://www.theguardian.com/world/2023/sep/01/greek-wildfires-a-visual-guide

Tiwari, S., Kumar, B., & Tyagi, A. (2022). Artificial Neural Network-based State of Charge (SOC) Estimation of a Lithium-Ion Battery under Different Temperatures Conditions. 2022 IEEE 10th Power India International Conference, PIICON 2022. https://doi.org/10.1109/PIICON56320.2022.10045116

US EPA. (2023). All EPA Emission Standards | US EPA. https://www.epa.gov/emission-standards-reference-guide/all-epa-emission-standards

Vidal, C., Kollmeyer, P., Chemali, E., & Emadi, A. (2019). Li-ion Battery State of Charge Estimation Using Long Short-Term Memory Recurrent Neural Network with Transfer Learning. ITEC 2019 - 2019 IEEE Transportation Electrification Conference and Expo. https://doi.org/10.1109/ITEC.2019.8790543

Yang, B. ;, Wang, Y. ;, Zhan, Y., Yang, B., Wang, Y., & Zhan, Y. (2022). Lithium Battery State-of-Charge Estimation Based on a Bayesian Optimization Bidirectional Long Short-Term Memory Neural Network. Energies 2022, Vol. 15, Page 4670, 15(13), 4670. https://doi.org/10.3390/EN15134670

Ząd ek, P., Koczor, A., Gołek, M., Matoga, Ł., & Penkala, P. (2015). Improving Efficiency of FPGA-in-the-Loop Verification Environment. IFAC-PapersOnLine, 48(4), 180–185. https://doi.org/10.1016/j.ifacol.2015.07.029