

Continual Reinforcement Learning using Real-World Data for Intelligent Prediction of SOC Consumption in Electric Vehicles

Juan P. Ortiz, *Member, IEEE*, Germán P. Ayabaca, *Member, IEEE* Angel R. Cárdenas, *Member, IEEE*
Diego Cabrera, *Member, IEEE* and Juan D. Valladolid, *Member, IEEE*

Abstract—The accelerated migration towards electric vehicles (EV) presents several problems to solve. The main aspect is the management and prediction of the state of charge (SOC) in real long-range routes of different variations in altitude for a more efficient energy consumption and vehicle recharge plan. This paper presents the implementation of a new algorithm for SOC estimation based on continuous learning and meta-experience replay (MER) with reservoir sample. It combines the Reptil meta-learning algorithm with the experience replay technique for stabilizing the reinforcement learning. The proposed algorithm considers several important factors for the prediction of the SOC in EV such as: speed, travel time, route altimetry, consumed battery capacity, regenerated battery capacity. A modified principal components analysis is used to reduce the dimensionality of the route altimetry data. The experimental results show an efficient estimation of the SOC values and a convergent increase in knowledge while the vehicle travels the routes.

Index Terms—Meta-experience replay (MER), reinforcement learning (RL), state of charge (SOC), electric vehicles (EV), neural networks (NN), principal component analysis (PCA).

I. INTRODUCCIÓN

Los Vehículos Eléctricos (VE) de batería captan la atención debido a factores ambientales, el uso de energías renovables y además poseen ventaja en emisiones contaminantes al compararlos con vehículos de combustión interna. Sin embargo, la aceptación de los VEs es limitada por la escasez de estaciones de carga y las limitantes en la autonomía de la batería [1]. Con el fin de contrarrestar esta problemática surgen propuestas para mejorar la autonomía del VE, de manera que pueda desplazarse de manera eficiente. En [2] plantea la relevancia del uso de baterías basado en la tecnología Li-ion, mencionando que la batería de Li-ion es la que ofrece mejores propiedades y actualmente es la más utilizada. Por otra parte, el análisis y optimización de parámetros de predicción de SOC aplicados en autobuses eléctricos, están basados en mediciones de distancia, aceleración y velocidad, concluyendo que se debe limitar la velocidad en el tráfico y evitar una aceleración innecesaria para eludir el frenado inmediato posterior.

J. P. Ortiz, G. P. Ayabaca, A. R. Cárdenas y J. D. Valladolid están con el Grupo de Investigación en Ingeniería del Transporte (GIIT) de la Universidad Politécnica Salesiana sede Cuenca, Ecuador. E-mail: jortizg@ups.edu.ec

D. Cabrera está con el Grupo de Investigación y Desarrollo en Tecnologías Industriales (GIDTEC) de la Universidad Politécnica Salesiana sede Cuenca, Ecuador. E-mail: dcabrera@ups.edu.ec

J. D. Valladolid también está con el Departamento de Ingeniería Electrónica de la Pontificia Universidad Javeriana, Bogotá, Colombia.

Las variables para la predicción del consumo de energía aparecen en una cantidad elevada, es decir, los algoritmos aplicados a este volumen de datos tienen que descubrir los atributos de las variables para realizar predicciones. Con este motivo sobresale una estrategia denominada reducción de dimensionalidad mediante el Análisis de Componentes Principales (PCA) [3].

La reducción de dimensionalidad identifica a partir de un conjunto de variables, otro conjunto nuevo de variables no directamente observables, denominadas factores que puedan expresar la información contenida en el conjunto original de los datos y reduce variables que son redundantes [4]. La clave consiste en reducir la dimensionalidad, sin sacrificar información importante. Los tres principales métodos de reducción de dimensionalidad se presentan en [5] estos son el PCA, el Análisis Discriminante Lineal (LDA) y el escalamiento multidimensional (MDS), siendo el PCA el único de los tres métodos mencionados que es no supervisado y reversible. Por ello, el presente artículo, considera el PCA como método de reducción de dimensionalidad porque facilita la interpretación y proporciona una mayor cantidad de información sobre la dinámica de los datos.

Por otro lado, el uso de Inteligencia Artificial (IA) está puesta en funcionamiento en varios campos de la industria automotriz [6]. Algo importante es saber que existen dos grupos de aprendizaje automático como son el aprendizaje supervisado y aprendizaje no supervisado [7]. La diferencia radica en que, para el aprendizaje supervisado los resultados que se esperan obtener son conocidos previamente; en cambio, en el aprendizaje no supervisado el resultado no se utiliza durante el entrenamiento de la Red Neuronal Artificial (RNA) [8]. Los sistemas de aprendizaje profundo son susceptibles a un fenómeno conocido como olvido catastrófico [9], ocurre cuando una red neuronal entrenada no es capaz de mantener su capacidad para realizar tareas previamente aprendidas [10]. La solución es el uso de aprendizaje continuo, donde el modelo observa los ejemplos relacionados con una secuencia de tareas y con la Memoria Episódica de Gradiente (GEM) mitigan el olvido de tareas aprendidas previamente, al tiempo que permite la transferencia del conocimiento de tareas anteriores [11], [12], [13].

El meta-aprendizaje, o aprender a aprender, es la ciencia de observar sistemáticamente cómo se desempeñan los diferentes enfoques de aprendizaje automático en una amplia gama de tareas de aprendizaje y luego aprender de esta experiencia

(meta-datos) con la finalidad de adaptarse a nuevas tareas mucho más rápido de lo que sería posible de otra manera [14]. Un problema de meta-aprendizaje particularmente desafiante es entrenar un modelo de aprendizaje profundo preciso utilizando solo unas pocas muestras de entrenamiento, de otra manera y dada la experiencia previa con tareas muy similares se necesitaría grandes conjuntos de entrenamiento disponibles. Los seres humanos tienen una capacidad innata para hacer esto y se desea crear agentes de aprendizaje automático que puedan hacer lo mismo [15].

Varios investigadores han definido el meta-aprendizaje de forma diferente según sus aplicaciones específicas [16]–[18]. Una categoría de meta-aprendizaje es el aprendizaje por conjuntos, que mejora la precisión del aprendizaje al combinar los resultados de varios aprendices de nivel base en forma paralela o secuencial [19]–[23]. La segunda categoría de meta-aprendizaje es la auto-adaptación/auto-modificación de parámetros. No requiere que el aprendiz-base y el meta-aprendiz sean diferentes. El meta-aprendiz son los algoritmos de aprendizaje que pueden autoadaptar parámetros o políticas del modelo [24]–[26]. El tercer tipo de meta-aprendizaje es la transferencia inductiva, que reutiliza el conocimiento adquirido previamente en tareas nuevas o similares para realizar un aprendizaje rápido, por ejemplo, aprendizaje de pocos datos (“few-shot learning”) [18], [27]–[32].

Asimismo, en el aprendizaje de pocos datos está bien documentado en el estado del arte el uso del algoritmo llamado “Reptil” [33]–[37]. Reptil es escalable y busca una inicialización para los parámetros de una red neuronal, de modo que la red se pueda ajustar utilizando una pequeña cantidad de datos y sin la necesidad de calcular una segunda derivada. La aproximación de los parámetros en cada tarea se hace mediante el cálculo del descenso de gradiente estocástico (SGD) de una manera estándar. Los resultados obtenidos en [38] muestran que el desempeño de Reptil es comparable al algoritmo de primer orden de meta-aprendizaje agnóstico del modelo (MAML) que es usado para la rápida adaptación de parámetros en redes neuronales profundas [33]. Todas estas características hacen que Reptil requiera menos cálculos y memoria, reduciendo de esta manera el coste computacional del algoritmo implementado y con una rápida adaptación.

En este trabajo, proponemos el uso del método de meta-aprendizaje continuo de pocos datos con repetición de la experiencia (MER) basado en el algoritmo de Reptil para la estimación del SOC de un VE. Las contribuciones de este paper son las siguientes: 1) El algoritmo MER implementado en conjunto con un reservorio de muestreo y una arquitectura de RNA de tres capas tipo feedforward puede adaptarse rápidamente con pocas muestras a varias condiciones de las rutas tanto en los tipos de calzada como en los modos de manejo, usando limitadas señales de entrada para la estimación del consumo del SOC. 2) La técnica de reservorio de muestreo mantiene sin saturación el almacenamiento de datos al no ser necesario tener todas las rutas del VE almacenadas para el aprendizaje y cálculo de la estimación. Además, se muestra en recorridos con condiciones reales que a pesar de tener una memoria limitada para los meta-datos el algoritmo evita el fenómeno de olvido catastrófico que es común en varios

de los sistemas de aprendizaje continuo. 3) Los experimentos realizados con datos obtenidos del VE muestran que el método propuesto tiene una rápida convergencia de la estimación y un error bajo mientras crece sin olvido el conocimiento de la red al realizar el recorrido de las nuevas rutas.

Las secciones que conforman el artículo se encuentran de la siguiente manera: en primer lugar, la Sección II recapitula la literatura relacionada y relevante para el desarrollo de este trabajo. La Sección III describe las variables que tendrán protagonismo para analizar el SOC junto con el desarrollo de la red neuronal. Enseguida, en la Sección IV se indica las diferentes variables entregadas por el VE en las rutas establecidas y los resultados de los experimentos. Finalmente, la Sección V presenta las conclusiones del trabajo.

II. TRABAJOS RELACIONADOS

A. Predicción y Estado de Carga de la Batería (SOC)

Varias investigaciones se han desarrollado en este campo para la estimación del SOC de baterías de iones de litio en un VE. Los algoritmos implementados son diversos como los basados en modelos, aprendizaje de máquina, redes neuronales y aprendizaje profundo. Entre los métodos de estimación basados en modelos se tiene el recuento de Coulomb (CC), el voltaje de circuito abierto (OCV), el filtro Kalman extendido (EKF), el filtro Kalman extendido adaptativo (AEKF) y el filtro Kalman Unscented (UKF) [39]–[43]. Algunos de los algoritmos basados en aprendizaje de máquina que se han usado son máquinas de soporte vectorial (SVM), regresiones lineales (LR), regresiones de procesos Gaussianos (GPR), ensemble bagging (EBa) y ensemble boosting (EBo) [44], [45]. Asimismo, varios modelos de redes neuronales se han probado para la estimación del SOC como redes neuronales feed-forward (RNN), redes neuronales independientes recurrentes (IndRNN), redes neuronales profundas (Deep Learning), entre otras [46].

La Tabla I muestra los resultados del error medio cuadrático (RMSE) de los algoritmos más relevantes mencionados con anterioridad. En este resumen, se consideran los resultados de investigaciones que cumplen con condiciones de experimentación similares para ser comparadas, estas son: una batería de 2.9Ah Panasonic 18650PF, temperatura ambiente de prueba de 25°C y los ciclos de carga/descarga [47]–[49].

Para el entrenamiento de estos métodos, los investigadores usaron bases de datos con una gran cantidad de datos llegando a ser inaplicables en nuestro caso particular. Sin embargo, los indicadores de desempeño RMSE permiten una comparación general de los resultados obtenidos por el algoritmo MER propuesto en este trabajo. En el estado del arte, no se ha identificado aún la aplicación de métodos de aprendizaje de refuerzo continuo para la estimación del SOC en condiciones de funcionamiento de ruta. Además, se considera que los datos de los recorridos en diferentes regiones geográficas son muy escasos y cambian constantemente. Por lo que, este artículo muestra una novedosa implementación con datos de recorridos en condiciones reales de un VE incluyendo diversos efectos que modifican el consumo del SOC como la altimetría de la ruta, el clima, el tipo de calzada; entre otras variables inherentes a su propio funcionamiento.

TABLA I
MÉTODOS DE ESTIMACIÓN DEL SOC [47]–[49]

Basados en modelo	
Método	RMSE [%]
CC with reseted Capacity	0.09
CC	2.11
OCV	4.51
EKF	14.33
AEKF	4.14
Luenberger	8.73
Aprendizaje de máquina	
Método	RMSE [%]
SVM	12.25
LR	3.6
GPR	4.1
Ensemble boosting	22.92
Ensemble bagging	4.11
Redes neuronales	
Método	RMSE [%]
RNN	0.61
IndRNN	0.76

B. Análisis de Componentes Principales (PCA)

El objetivo de los PCA es descomponer las variables que están conectadas para reducir al mínimo el número de datos, manteniendo la información más relevante para mejorar el análisis [50], logrando visualizar datos en dos dimensiones. Por esta razón se realiza un ajuste de datos de las diferentes muestras con $\mathbf{X} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{n \times s}$, (donde n es el número de muestras y s es el número de variables) que se representa en la siguiente ecuación:

$$\mathbf{X}'(t) = \frac{\mathbf{X}(t) - \bar{\mathbf{X}}}{D(\mathbf{X})} \quad (1)$$

Donde $\bar{\mathbf{X}}$ representa la media de los datos, $D(\mathbf{X})$ es la desviación estándar y t denota el tiempo de muestreo [51].

Un comportamiento particular del cálculo de PCA se da cuando al conjunto de datos de entrada se agrega una nueva ruta, esta adición provoca que los puntos en el subespacio de dimensiones reducidas (PC_1 vs. PC_2) roten cambiando las posiciones de sus coordenadas. Por ello, se recalcula la rotación como se observa en la Fig. 1, donde el punto p_1 es ($C = (x_1; y_1)$), de la misma forma el punto p_2 es ($D = (x_2; y_2)$) siendo parte del segmento de recta P_2 . Para el segmento de recta P_1 se tiene dos puntos: el punto p_3 es ($A = (fx_1; fy_1)$) y el punto p_4 es ($B = (fx_2; fy_2)$). A continuación, se determina f y g que son los puntos medios existentes entre los vectores de P_1 y P_2 respectivamente. Se forma un vértice con un ángulo de 90° entre las rectas de los puntos medios calculados. Entonces se calcula las dos pendientes posibles, eligiendo a f en primer lugar y luego a g , con la ecuación:

$$m_f = \frac{(y_2 - y_1)}{(x_2 - x_1)} \quad (2)$$

Con las pendientes calculadas y los puntos medios (f y g) se obtienen dos rectas que se intersectan en el punto E siendo este la posición del centro de rotación para todos los puntos del plano. El ángulo de rotación se calcula mediante la resta

de los ángulos que se forman con respecto a la horizontal de los vectores k y l como se observan en la Fig. 1. Finalmente, con los datos del centro y el ángulo de rotación, se tienen las muestras en sus posiciones originales y la nueva ruta ya no está sesgando la distribución inicial. Este proceso de rotación se debe realizar cada vez que se incorpore una nueva ruta.

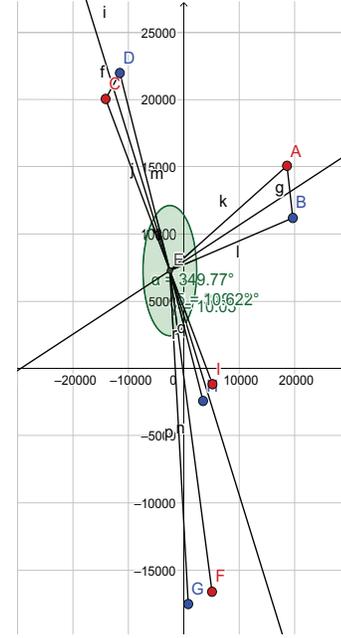


Fig. 1. Análisis geométrico para la rotación de las coordenadas de PCA. El punto G es el PCA original (Azul) de la ruta actual y el punto F es el PCA rotado (Rojo) de la ruta actual.

III. FORMULACIÓN DEL PROBLEMA

A. Análisis del Vehículo Eléctrico

Los factores que influyen de manera directa en la descarga de la batería del VE son: distancia recorrida [km], tiempo [min], capacidad de descarga y carga de la batería [Ah] [52]–[54]. El SOC [%] se define como la relación de la carga restante dentro de la batería expresada por:

$$SOC[\%] = \frac{Q_r}{Q_N} 100\% \quad (3)$$

Donde Q_N representa la capacidad nominal, Q_r representa la capacidad residual. Los métodos tradicionales de estimación de SOC son los métodos de conteo de Coulomb y voltaje de circuito abierto. El método de conteo de Coulomb integra la corriente en el tiempo y se representa matemáticamente de la siguiente forma:

$$SOC(t_1) = SOC(t_0) - \int_{t_0}^{t_1} \eta I(t) dt / Q_N \quad (4)$$

En que $SOC(t_0)$ es el valor del SOC en el tiempo inicial t_0 , η es la eficiencia coulombica, $I(t)$ es la corriente de la batería con un valor positivo en la descarga y un valor negativo en la carga y Q_N es la capacidad nominal de la batería [55]. De otra forma [53], hace uso de sensores virtuales para la estimación de la corriente de la batería y considera factores que están disponibles en el vehículo como: voltaje de la batería,

TABLA II
BASE DE DATOS RUTAS RECORRIDAS POR EL VE

Num.	Ruta	V. Media. [km/h]	Distancia [km]	Tiempo [min]	SOC [%]	Carga [Ah]	Descarga [Ah]
0	UPS - AZOGUES	55,78	29	31	13,5	3,4	13,9
1	AZOGUES - UPS	65,3	28,6	26	14	4	15,3
2	UPS - CAJAS TRES CRUCES	46,46	37,4	47	41	3,5	35,9
3	CAJAS TRES CRUCES - UPS	58,02	35,7	36	7,5	13,6	7,7
4	UPS - YUNGUILLA	56,15	79,3	84	18,5	26,4	41,5
5	YUNGUILLA - UPS	45,49	74,6	97	55,5	13,4	57,3
6	TURI - BANOS	37,04	7,9	12	4	2,2	5,6
7	UPS - SAN ANDRES JADAN	40,69	26,4	38	13	3,6	14,1
8	SAN ANDES JADAN - UPS	43,63	26,8	36	10	6,2	14
9	PARQUE CALDERON - MILCHICHIG	20,27	6,2	17	1	1,2	2
10	UPS - CENTRO MACHALA	57,54	185,7	266	63,5	59,7	112,2
11	UPS - GUALALCAY_VALLE	36,39	16,6	27	14,5	1,9	13,4
12	GUALALCAY_VALLE - UPS	37,09	17,1	27	4,5	3,3	6,9
13	UPS - COJITAMBO	49,42	29,1	36	23,12	1,7	20,2
14	COJITAMBO - UPS	51,95	26,3	30	7	4,4	9,7
15	UPS - TUTUPALI	27,03	19,3	42	16,5	1,9	15,1
16	TUTUPALI - UPS	38,02	25,1	39	3,5	6,3	9,3
17	UPS - GUALACEO	44,62	34,7	46	9	3	10,3
18	GUALACEO - SIGSIG	39,04	25	37	16,5	2,3	15,3
19	GUALACEO - UPS	48,28	33,9	41	18,5	2,2	16,6
20	MACHALA - S. ISABEL	54,3	93,6	102	76	8,6	68,9
21	UPS - CAÑAR	55,85	63,6	70	42,5	7,8	41,5
22	CAÑAR - UPS	68,66	63,3	54	21	14,6	31,3
23	M. SOL - TAMARINDO	64,89	86	78	55,5	7,8	51,7
24	UPS - P. CRISTO CONSUELO	27,69	15,6	33	14	1,2	12,3
25	P. CRISTO CONSUELO - UPS	23,99	15	36	3,5	3,8	6,4
26	UPS - GULLANZHAPA	41,72	22,6	32	17,5	1,7	15,8
27	GULLANZHAPA - UPS	41,95	22,7	32	4,5	4,3	7,6
28	UPS - AGUARONGO	34,6	32,34	55	31	2,8	27,3
29	AGUARONGO - UPS	31,63	32,8	61	8,5	6,9	13,8

corriente de la batería, el SOC de la batería, posición del pedal y la velocidad del vehículo, teniendo como resultado de funcionamiento del sensor virtual y el modelo de batería que permite estimar el estado de carga con un ajuste superior al 87%.

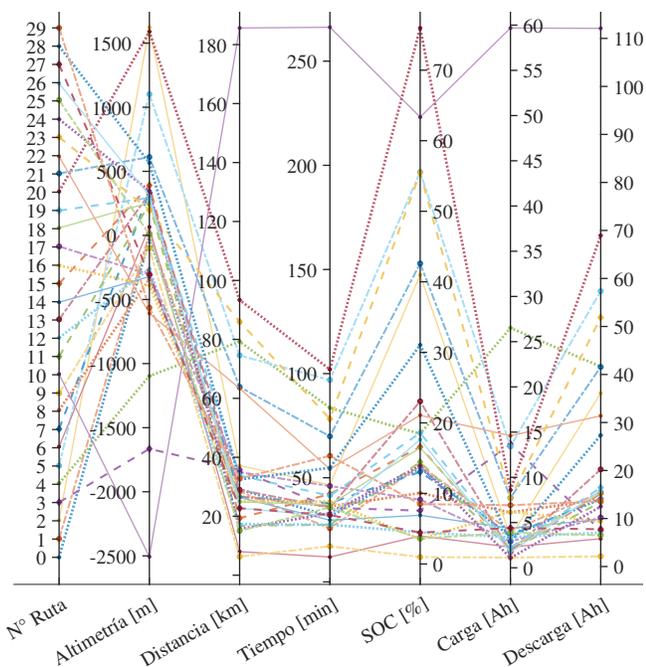


Fig. 2. Representación de los factores considerados en Coordenadas Paralelas.

En la Fig. 2, se observa la relación entre los factores definidos en la Tabla II. Los resultados muestran la variación de altimetría de las rutas con una pendiente positiva de 1598 metros sobre el nivel del mar (ms.n.m.) cuando la ruta es de Costa a la Sierra, mientras que cuando las rutas son de Sierra a la Costa, se tiene una pendiente negativa de 2500ms.n.m. En cuanto a la distancia recorrida, el sistema consta con una ruta de 185.7km de recorrido aunque la gran mayoría de rutas van de los 15 a 37km, ya que son rutas aleañas a la ciudad de Cuenca. Por último, se tiene el consumo máximo con descarga de 112.2Ah y la regeneración máxima con valores de carga de 59.7Ah, esto debido a la carga regenerativa del vehículo eléctrico, ya que la trayectoria de la ruta es de pendiente negativa (UPS-CENTRO MACHALA) mostrada en la Tabla II.

B. Ejecución del Algoritmo de Meta-Aprendizaje

La Fig. 3 describe los pasos para la estimación del consumo del SOC. En el inicio se introduce el origen y el destino de una ruta, luego se consulta a Google Maps los datos de altimetría, tiempo y distancia de la ruta ingresada. Después, la dimensionalidad de la altimetría es reducida mediante PCA para que se formen dos archivos, el primero con las coordenadas de PCA sin la altimetría de la ruta nueva y el segundo archivo con las coordenadas de PCA con la altimetría de una nueva ruta y la de las rutas anteriores. Después de tener los valores, se aplica la RNA para obtener la estimación del consumo del SOC y mostrar los resultados al usuario. Posteriormente, se procede a conducir por la ruta trazada y enseguida se almacena

la información del VE a lo largo del trayecto. Cuando el recorrido finaliza, es necesario acceder a los datos del VE, los mismos que son: SOC consumido, energía consumida y energía regenerada. Estos datos tienen que ser comparados con los datos de la estimación para calcular el error existente. En el caso de que el error sea mayor al 1%, tiene que reentrenarse la RNA usando el Aprendizaje Reforzado Continuo con MER (Algoritmo 2), o en el caso de que el error sea menor al 1%, se puede reiniciar todo el proceso, es decir, se consulta al usuario si desea predecir el consumo de SOC de una nueva ruta, o si desea finalizar el proceso.

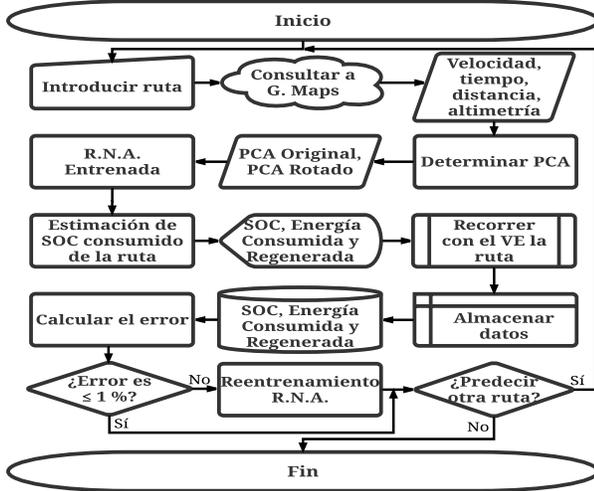


Fig. 3. Diagrama de flujo del algoritmo de estimación.

C. Aprendizaje Reforzado Continuo con Meta-Experience Replay (MER)

En este trabajo se considera la modificación del algoritmo de Reptil presentado en [12] que se fusiona correctamente con un módulo de repetición de experiencia (Algoritmo 2), facilitando el aprendizaje continuo al maximizar la transferencia y minimizar la interferencia. El módulo de reserva de memoria (Algoritmo 1) contiene un espacio de memoria N suficiente para mantener las características del evento y también para disminuir el costo computacional del entrenamiento de la red neuronal. El data set $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ de n muestras, contiene un vector de características de entrada $\mathbf{x}_i \in X$ y un vector de objetivos $\mathbf{y}_i \in Y$. La función de aproximación es $f_\theta(\mathbf{x}) : X \rightarrow Y$ donde θ corresponde a los parámetros estimados de los datos que en este caso son los pesos de una red neuronal. Dando una función de costo $L(f_\theta(\mathbf{x}), \mathbf{y})$, la estimación de parámetros se formula como un problema empírico de minimización de riesgos:

$$\min_{\theta} \frac{1}{|D|} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \sim D} L(f_\theta(\mathbf{x}), \mathbf{y}) \quad (5)$$

La configuración comúnmente usada para definir condiciones no-estacionarias para aprendizaje continuo se muestra en [11], donde cada tarea t_k es un problema de aprendizaje supervisado asociado con sus propios datos de distribución de probabilidad

$P_k(\mathbf{x}, \mathbf{y})$. En consiguiente, se tiene una secuencia (potencialmente infinita) como sigue:

$$(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_i, \mathbf{y}_i), \dots, (\mathbf{x}_{i+j}, \mathbf{y}_{i+j})$$

Algoritmo 1 Reservorio de Muestreo con Algoritmo R

```

RESERVOIR( $M, N, \mathbf{x}, \mathbf{y}$ )
if  $M < N$  then
   $M[N] \leftarrow \{\mathbf{x}, \mathbf{y}\}$ 
else
   $i = \text{randomInteger}(\text{min} = 0, \text{max} = N)$ 
  if  $i < M$  then
     $M[i] \leftarrow \{\mathbf{x}, \mathbf{y}\}$ 
  end if
end if
  
```

Algoritmo 2 Meta-Aprendizaje de Repetición con Reservorio de Muestreo

```

TRAIN( $D, \theta, \alpha, \gamma, sk, N_{iter}$ )
 $M \leftarrow \{\}$ 
for  $t = 0, 1, \dots, T$  do
  for  $\mathbf{x}$  in  $D_t$  do
    for  $iter = 0, 1, \dots, N_{iter}$  do
       $\theta_0 \leftarrow \theta$ 
      for  $i = 1, \dots, sk$  do
         $B_i \leftarrow \text{sample}(\mathbf{x}, \mathbf{y}, sk, M)$ 
         $\theta_i \leftarrow \text{SDG}(B_i, \theta_{i-1}, \alpha)$ 
      end for
      // Linear schedule :
       $\gamma_c \leftarrow \gamma(1 - \frac{iter}{N_{iter}})$ 
      // Reptile meta - update :
       $\theta \leftarrow \theta_0 + \gamma_c(\theta_{sk} - \theta_0)$ 
    end for
    // Reservoir sampling memory update :
     $M \leftarrow M \cup \{\mathbf{x}, \mathbf{y}\}$ 
  end for
end for
  
```

MER mantiene una memoria de estilo de repetición de experiencias M con reservorio de muestras y en cada iteración de tiempo extrae lotes que incluyen $sk - 1$ muestras aleatorias del búfer para ser entrenadas junto con la muestra actual. El algoritmo de Reptil se implementa optimizando a través de sk lotes de datos secuenciales con un optimizador basado en el SGD y una tasa de aprendizaje α . Después de entrenar en estos lotes, se toman los parámetros iniciales antes de entrenar θ_0 y se actualizan a $\theta \leftarrow \theta_0 + \gamma_c(\theta_{sk} - \theta_0)$ donde γ_c es la tasa de aprendizaje para la actualización de meta-aprendizaje. El proceso se repite para cada serie de lotes sk hasta llegar a un número máximo de iteraciones N_{iter} (Algoritmo 2).

La función de pérdida de aprendizaje típica tiene una ponderación proporcional a sk y el término regularizador para maximizar la transferencia y minimizar la interferencia tiene una ponderación proporcional a $\gamma_c \leftarrow \gamma(1 - \frac{iter}{N_{iter}})$. Esto implica que al maximizar los valores efectivos se puede dar más peso al término de regularización. Para un número fijo de

muestras extraídas de M , se obtiene un mejor rendimiento consistente en la conversión a una larga lista de muestras individuales que con el uso de lotes adecuados para un aprendizaje de pocas iteraciones como se presenta en [33].

IV. ANÁLISIS DE RESULTADOS

En el desarrollo del artículo se define un objetivo que es conocer la autonomía de la batería del VE aplicando un algoritmo MER. Esta solución se basa en la reducción de dimensiones mediante PCA, extrayendo parámetros definidos en la Tabla III. Mediante esta información se predice el consumo de energía del VE en rutas previamente establecidas, cuyos datos se obtienen mediante el Bus CAN y el protocolo OBD-II del VE. En [56] y [57], se muestra cómo es transferida la información de las variables mediante un software de adquisición de datos. El conjunto de datos de altimetría, influye principalmente en el consumo de la energía; esta información se obtiene de la API key de la plataforma de Google Maps que ofrece varios datos como: la elevación de una ruta, la extracción de coordenadas GPS en tiempo real, la distancia recorrida, etc.

Para esto se desarrolló una aplicación en lenguaje de programación Python para extraer los valores de altimetría. Más tarde se determina el PCA de la ruta, para predecir el consumo y la regeneración de la batería. El entrenamiento de la RNA con los datos del PCA adquiridos se basa en la repetición de la experiencia con el algoritmo de entrenamiento Reptil. La Fig. 4, muestra el diagrama de bloques de los instrumentos necesarios para desarrollar este artículo.

El VE usado para la experimentación en este trabajo es un modelo KIA SOUL. Las características en el sistema de batería de alto voltaje son: baterías de polímero de Li-ion, posee 96 celdas en conexión serie, un voltaje nominal de 370v y una capacidad de 80Ah.

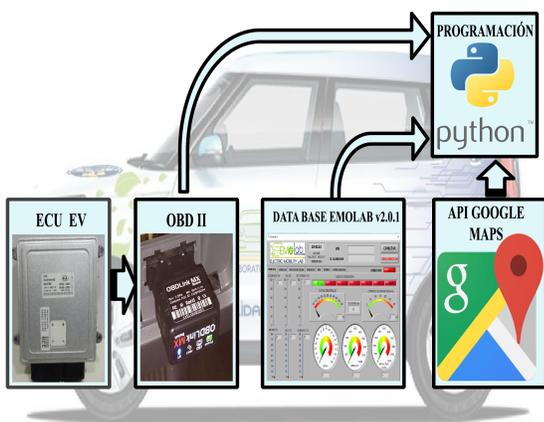


Fig. 4. Diagrama de bloques del sistema de adquisición de datos para el VE.

A. Altimetría

El estudio es realizado en la ciudad de Cuenca-Ecuador, con 2560 ms.n.m. En la Fig. 5, se muestra la altimetría de 6 rutas, por ejemplo, la ruta No. 2 tiene origen en la Universidad

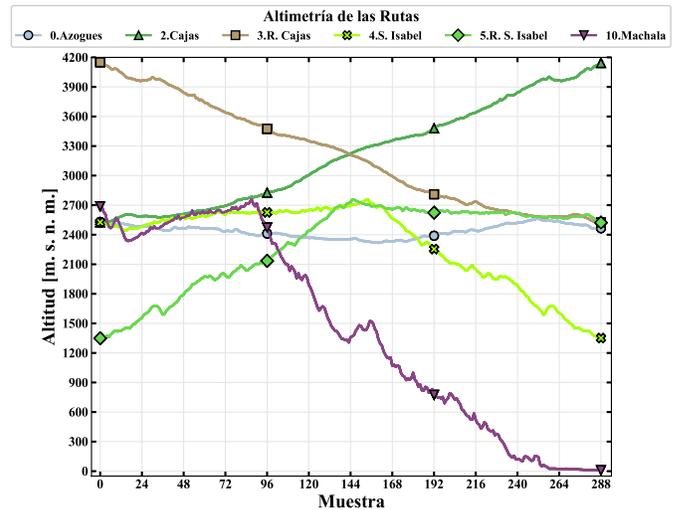


Fig. 5. Rutas de referencia con mayor variación de altimetría.

Politécnica Salesiana – Sede Cuenca y con destino hacia el Parque Nacional El Cajas - Mirador Tres Cruces.

Mediante el puerto de diagnóstico se adquiere la información necesaria del VE, el tiempo de muestreo de los datos es de 1 s para cada ruta planteada en Google Maps como se observa en la Fig. 6. Durante el recorrido de una ruta se forman

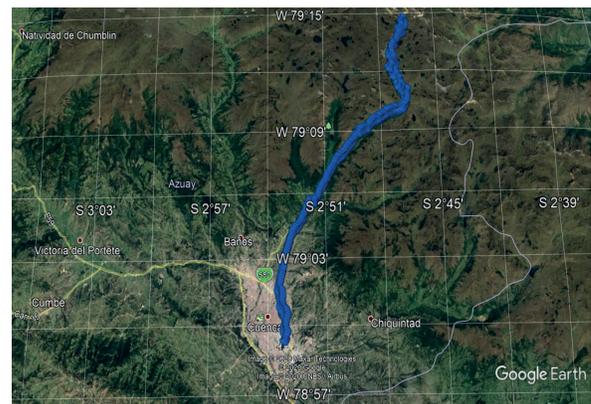


Fig. 6. Ruta: UPS - Cajas, con 29 [km] de distancia, 31 [min] de tiempo, trazada en Google Earth para verificar el recorrido.

un conjunto de valores, que son una combinación de datos de GPS del VE, información de la carretera por segmento y altitud con 290 muestras en toda la ruta. Los datos registrados al final de cada ruta se observan en la Tabla II. A continuación, se procede a la reducción de dimensiones de los datos de la altimetría mediante PCA. Dado que los datos originales de PCA deben pasar por un proceso de rotación para mantener su posición al momento de agregar una nueva ruta se utiliza el método presentado en la sección B, de esta forma, se consigue datos en dos dimensiones como se observa en la Tabla III contando con nuevos datos llamados PCA Rotado. Para el procedimiento de rotación de PCA se consideraron 11 rutas (las 10 primeras rutas mostradas en la Tabla II más la ruta actual). En la Fig. 7, se representa varias características de la altimetría de las 30 rutas, en un plano más compacto aplicando el PCA para que la RNA no procese información excesiva y

TABLA III
ROTACIÓN DE LOS PC

Rutas	PCA Original		PCA Rotado	
	PC1	PC2	PC1	PC2
0	28,319	-655,891	28,147	-655,877
1	-683,426	-13,300	-683,717	-13,253
2	-15677,108	-715,215	-15677,108	-715,215
3	16842,593	-689,239	16843,407	-689,446
4	4955,699	5285,095	4956,702	5284,846
5	-18152,811	6360,711	-18152,367	6360,614
6	-280,947	-233,238	-280,738	-233,300
7	-691,934	-819,454	-691,855	-819,484
8	2294,820	-1266,625	2294,947	-1266,675
9	-237,624	570,141	-237,461	570,086
10	25211,337	2830,437	25210,702	2830,538
11	-2828,940	-123,024	-2828,895	-123,049
12	2108,689	1,226	2108,836	1,174
13	-543,854	-1141,838	-543,772	-1141,864
14	4820,304	-2914,421	4821,001	-2914,596
15	-3318,204	227,251	-3318,177	227,239
16	4503,024	-1152,785	4503,828	-1152,980
17	2020,008	219,245	2020,322	219,156
18	-3326,910	555,006	-3327,149	555,043
19	-3666,198	476,470	-3666,097	476,435
20	-13304,401	-1509,104	-13304,102	-1509,161
21	-8201,219	-3069,339	-8202,025	-3069,201
22	6665,877	1961,857	6664,346	1962,173
23	-1398,946	489,654	-1398,670	489,581
24	-5002,189	93,179	-5001,974	93,118
25	3293,399	749,493	3293,644	749,420
26	-3815,332	-679,617	-3815,239	-679,654
27	3072,184	101,546	3072,218	101,520
28	-2507,985	-1837,189	-2508,019	-1837,189
29	7821,777	-3101,032	7822,383	-3101,187

sea más eficaz, ahorrando de esta manera recursos y tiempo para su entrenamiento.

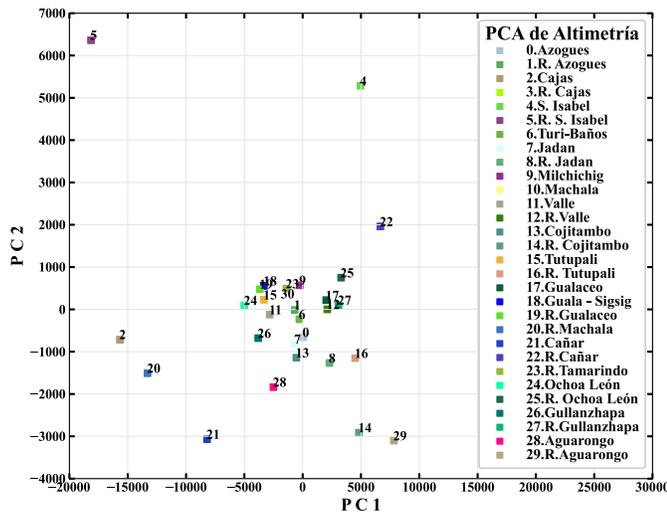


Fig. 7. Reducción de la dimensionalidad aplicando el PCA a la altimetría de las rutas.

B. Predicción y Resultados

Para el entrenamiento de la RNA, se definen 4 entradas que son las variables de: tiempo, distancia, componentes principales PC_1 y PC_2 ; ilustradas en las Tablas II y III. La RNA posee 2 capas ocultas compuestas por 50 neuronas con funciones de activación tangente hiperbólica. En la capa de salida se aplica la función de activación ReLu propuesta en

[58]. La red neuronal entrega la información de la energía consumida y la energía regenerada del VE. En la Fig. 8, se aprecia la arquitectura de la red neuronal para la predicción de consumo de batería del VE. La red tiene una tasa de aprendizaje igual a 0.005, los parámetros de la red considerados para el Algoritmo 1 y Algoritmo 2 son: $\gamma = 0.9$, $sk = 2$, $N_{iter} = 100$ y $N = 30$, respectivamente.

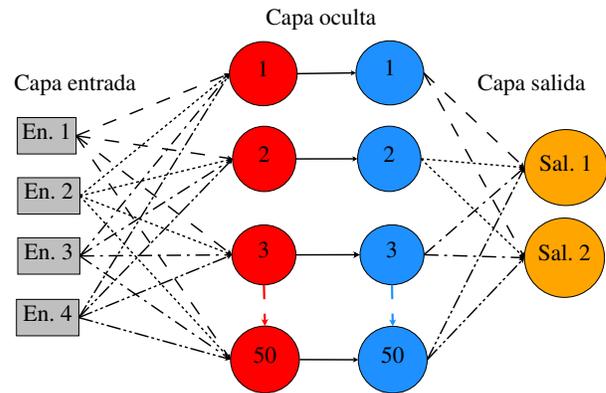


Fig. 8. Diagrama general de la red neuronal.

En la Fig. 9, se observa que la estimación en rutas de corto recorrido tiene una alta similitud. En rutas de largo recorrido existe una desviación entre los valores reales y estimados. Por ejemplo, la ruta No. 22 tiene una distancia de 63.3km con un valor de predicción de carga de 17.77Ah, mientras que, en la prueba de recorrido el valor real de carga fue 14.6Ah. Esta variación se debe principalmente a la diferencia en la velocidad de conducción.

Por otro lado, en la Fig. 10 se aprecia que desde la ruta No. 11 hasta la ruta No. 28, los valores son ampliamente similares. Sin embargo, para la ruta No. 29 el valor estimado es 12.57Ah y el valor real del recorrido es de 13.2Ah. Un aspecto a considerar en esta variación es el tipo de vía, debido a que no fue de asfalto o pavimento como las demás rutas.

Con los valores estimados de capacidad de carga y descarga, es posible calcular el SOC que consumiría el VE. Para

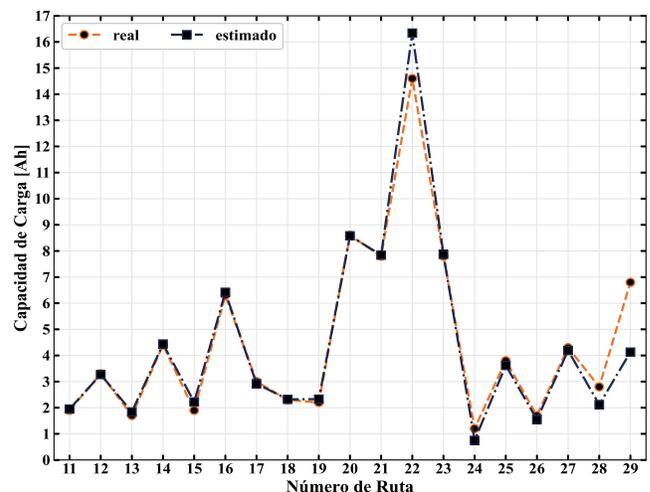


Fig. 9. Comparación de los valores de capacidad de carga de corriente reales y estimados.

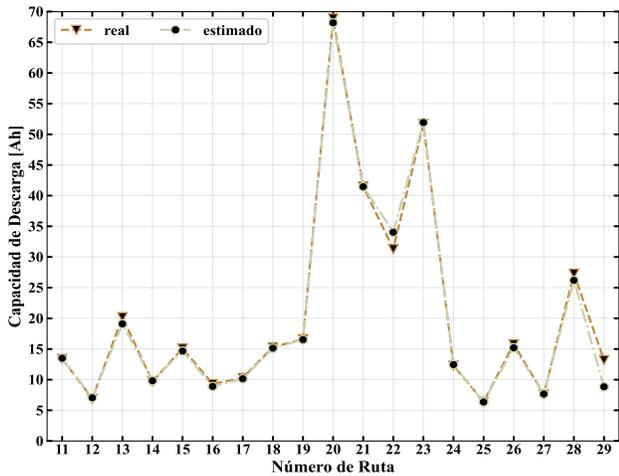


Fig. 10. Comparación de los valores de capacidad de descarga de corriente reales y estimados.

determinar el SOC, se utiliza la siguiente ecuación:

$$SOC[\%] = \frac{Descarga[Ah] - Carga[Ah]}{80[Ah]} \times 100[\%] \quad (6)$$

Los valores calculados con la ecuación (6) y los valores medidos del SOC de la ruta No. 11 hasta la ruta No. 29, se representan en Fig. 11. Se observa que los valores estimados del SOC son altamente semejantes a los valores reales. Se

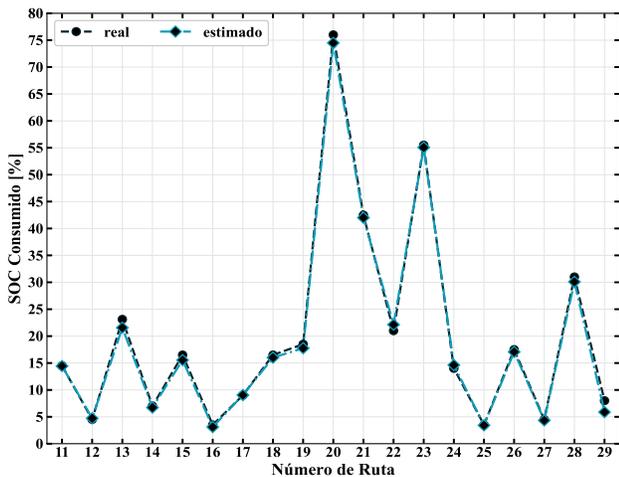


Fig. 11. Comparación de los valores del SOC consumido por el VE y el SOC calculado a partir de la estimación realizada por la red neuronal.

determinó los índices de desempeño de los valores estimados de la red neuronal y los valores reales, haciendo uso de los índices MAE (Error Absoluto Medio) y RMSE (Raíz del Error Cuadrático Medio), expresados en las ecuaciones (7) y (8).

$$MAE = \frac{1}{n} * \sum_{i=1}^n |V_{real} - V_{estimado}| \quad (7)$$

$$RMSE = \sqrt{\frac{1}{n} * \sum_{i=1}^n (V_{real} - V_{estimado})^2} \quad (8)$$

En la Fig. 12, se representa los resultados de MAE y RMSE obtenidos de las 29 rutas. Cuando se tiene los primeros datos, en la ruta No. 13 por ejemplo, el error es mayor, el SOC

estimado fue de 21.56%, pero el SOC consumido por el VE fue de 23.12%, la posible causa sea el tiempo de recorrido, por qué existe una diferencia de 7 minutos más entre el tiempo que se recorre con el VE y el tiempo de Google Maps, además de tener una altimetría diferente a la de las primeras rutas.

En el caso de la ruta No. 19, el error es menor debido a que las rutas No. 17 y No. 18, comparten similares características de altimetría. Es decir, el sistema ya posee un conocimiento previo para la estimación de esta ruta. De igual forma sucede para las rutas No. 24, No. 25, No. 26 y No. 27.

Otros casos de análisis son las rutas No. 22 y No. 29, el error nuevamente es mayor en ambos casos, las posibles causas pueden deberse a factores como el clima nublado y lluvioso. Además, el tiempo de recorrido en la ruta No. 22 se cumple con 15 minutos menos en el VE vs. el tiempo de Google Maps. En ambos casos se tienen modos de conducción agresiva, lo que provocaría los errores vistos. Al mismo tiempo, la ruta No. 29 se encontraba en muy mal estado con una constitución mixta de 40% de pavimento y 60% de tierra, esto provocó que durante la conducción se use con mayor frecuencia los pedales de freno y del acelerador del VE, resultando en consecuencia las desviaciones en los valores mostrados en las Fig. 9 y Fig. 10 para esta ruta. A pesar de ello, se puede observar que el cálculo del SOC consumido (Ecuación 6) basado en estas dos variables estimadas de la red neuronal entrega una buena aproximación con apenas el 1.5% de error absoluto (Fig. 11). Los indicadores máximos obtenidos son de 0.91% para el MAE y 0.68% para el RMSE (Fig. 12), siendo el algoritmo MER implementado bastante más competitivo en comparación con otros algoritmos de predicción de SOC presentados en la Tabla I, incluso al trabajar con pocas rutas como es este caso.

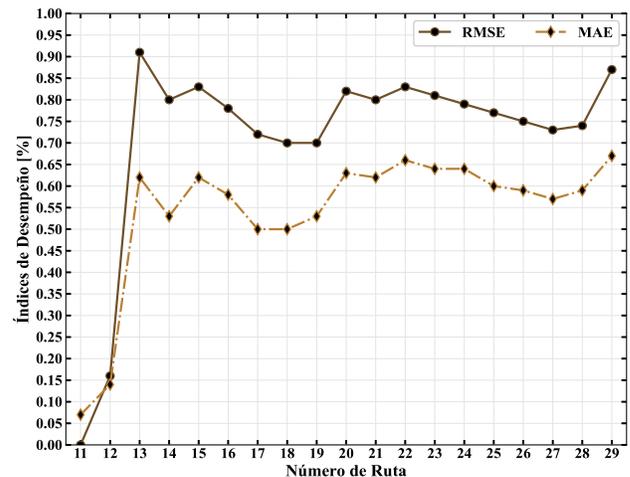


Fig. 12. Índices de Desempeño RMSE y MAE.

V. CONCLUSIONES

La contribución principal de este trabajo, es la implementación del algoritmo de aprendizaje reforzado continuo MER en un vehículo eléctrico para la predicción del SOC en rutas reales considerando la variabilidad en su altimetría.

Con base a los resultados de la experimentación se puede observar que el algoritmo propuesto converge rápidamente a los datos reales de consumo de SOC obteniéndose indicadores

máximos de desempeño MAE y RMSE menores al 1%. La técnica de almacenamiento de las rutas usada en el algoritmo 1 permite manejar una memoria y carga computacional limitada sin afectar el eficiente entrenamiento reforzado continuo de los datos basado en el algoritmo Reptil de meta-aprendizaje. Además, esta arquitectura permite que los tiempos de entrenamiento no se incrementen con el aumento del número de rutas recorridas, ya que la base de datos de entrenamiento en el algoritmo se restringe a un valor máximo de 30 muestras manteniendo sin olvido del conocimiento la información de todas las rutas.

RECONOCIMIENTO

Este estudio fue soportado por el Grupo de Investigación en Ingeniería del Transporte (GIIT) a través del proyecto de investigación Movilidad Verde de la Universidad Politécnica Salesiana sede Cuenca-Ecuador.

REFERENCES

- [1] V. T. Tran, D. Sutanto, and K. M. Muttaqi, "The state of the art of battery charging infrastructure for electrical vehicles: Topologies, power control strategies, and future trend," in *2017 Australasian Universities Power Engineering Conference (AUPEC)*, Nov 2017, pp. 1–6.
- [2] S. Zhang, R. Zhao, J. Gu, and J. Liu, "A numerical study of lithium-ion battery fast charging behaviors," in *2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, Aug 2018, pp. 449–454.
- [3] Y. M. Yeap and D. Tran, "Analysis of driving behavior's impact on battery discharge rate for electric vehicles," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 474–479.
- [4] L. Wang, "Research on distributed parallel dimensionality reduction algorithm based on pca algorithm," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, March 2019, pp. 1363–1367.
- [5] M. Lamrini and M. Y. Chkouri, "Decomposition and visualization of high-dimensional data in a two dimensional interface," in *2019 1st International Conference on Smart Systems and Data Science (ICSSD)*, Oct 2019, pp. 1–5.
- [6] J. Liu, X. Kong, F. Xia, X. Bai, L. Wang, Q. Qing, and I. Lee, "Artificial intelligence in the 21st century," *IEEE Access*, vol. 6, pp. 34 403–34 421, 2018.
- [7] O. Kotlyar, M. Pankratova, M. Kamalian, A. Vasychenkova, J. E. Prilepsky, and S. K. Turitsyn, "Unsupervised and supervised machine learning for performance improvement of nft optical transmission," in *2018 IEEE British and Irish Conference on Optics and Photonics (BICOP)*, Dec 2018, pp. 1–4.
- [8] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, April 2017.
- [9] C. Kaplanis, M. Shanahan, and C. Clopath, "Continual reinforcement learning with complex synapses," *arXiv preprint arXiv:1802.07239*, 2018.
- [10] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong gan: Continual learning for conditional image generation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019, pp. 2759–2768.
- [11] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 6467–6476.
- [12] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," *arXiv preprint arXiv:1810.11910*, pp. 1–31, 2019.
- [13] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 348–358.
- [14] J. Vanschoren, "Meta-learning: A survey," *arXiv preprint arXiv:1810.03548*, 2018.
- [15] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, 2017.
- [16] P. Brazdil, C. G. Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to data mining*. Springer Science & Business Media, 2008.
- [17] S. J. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan, and P. K. Chan, "Jam: Java agents for meta-learning over distributed databases," in *KDD*, vol. 97, 1997, pp. 74–81.
- [18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.
- [19] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
- [20] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [21] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [22] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [23] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine learning*, vol. 36, no. 1, pp. 105–139, 1999.
- [24] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.
- [25] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, 2016, pp. 3981–3989.
- [26] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.
- [27] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.
- [28] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International conference on machine learning*. PMLR, 2016, pp. 1842–1850.
- [29] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, pp. 3630–3638, 2016.
- [30] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.
- [31] F. Zhou, B. Wu, and Z. Li, "Deep meta-learning: Learning to learn in the concept space," *arXiv preprint arXiv:1802.03596*, 2018.
- [32] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3018–3027.
- [33] A. Nichol and J. Schulman, "Reptile: a scalable metalearning algorithm," *arXiv preprint arXiv:1803.02999*, vol. 2, p. 2, 2018.
- [34] Y. Tian and P. J. Gorinski, "Improving end-to-end speech-to-intent classification with reptile," *arXiv preprint arXiv:2008.01994*, 2020.
- [35] C. Tian, X. Zhu, Z. Hu, and J. Ma, "A transfer approach with attention reptile method and long-term generation mechanism for few-shot traffic prediction," *Neurocomputing*, vol. 452, pp. 15–27, 2021.
- [36] R. Singh, V. Bharti, V. Purohit, A. Kumar, A. K. Singh, and S. K. Singh, "Metamed: Few-shot medical image classification using gradient-based meta-learning," *Pattern Recognition*, p. 108111, 2021.
- [37] X. Liu, X. Zhang, W. Peng, W. Zhou, and W. Yao, "A novel meta-learning initialization method for physics-informed neural networks," *arXiv preprint arXiv:2107.10991*, 2021.
- [38] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.
- [39] R. Zhang, B. Xia, B. Li, L. Cao, Y. Lai, W. Zheng, H. Wang, and W. Wang, "State of the art of lithium-ion battery soc estimation for electrical vehicles," *Energies*, vol. 11, no. 7, p. 1820, 2018.
- [40] I. Baccouche, S. Jemmal, A. Mlayah, B. Manai, and N. E. B. Amara, "Implementation of an improved coulomb-counting algorithm based on a piecewise soc-ocv relationship for soc estimation of li-ionbattery," *arXiv preprint arXiv:1803.10654*, 2018.
- [41] Z. Chen, S. Qiu, M. A. Masrur, and Y. L. Murphey, "Battery state of charge estimation based on a combined model of extended kalman filter and neural networks," in *The 2011 International Joint Conference on Neural Networks*. IEEE, 2011, pp. 2156–2163.

- [42] W. Wang, X. Wang, C. Xiang, C. Wei, and Y. Zhao, "Unscented kalman filter-based battery soc estimation and peak power prediction method for power distribution of hybrid electric vehicles," *IEEE Access*, vol. 6, pp. 35 957–35 965, 2018.
- [43] H. Tian and B. Ouyang, "Estimation of ev battery soc based on kf dynamic neural network with ga," in *2018 Chinese Control And Decision Conference (CCDC)*, June 2018, pp. 2720–2724.
- [44] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, and K. Dietmayer, "Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods," *Journal of power sources*, vol. 239, pp. 680–688, 2013.
- [45] K. Liu, X. Hu, Z. Wei, Y. Li, and Y. Jiang, "Modified gaussian process regression models for cyclic capacity prediction of lithium-ion batteries," *IEEE Transactions on Transportation Electrification*, vol. 5, no. 4, pp. 1225–1236, 2019.
- [46] S. Song, Z. Wei, H. Xia, M. Cen, and C. Cai, "State-of-charge (soc) estimation using t-s fuzzy neural network for lithium iron phosphate battery," in *2018 26th International Conference on Systems Engineering (ICSEng)*, Dec 2018, pp. 1–5.
- [47] A. Manthopoulos and X. Wang, "A review and comparison of lithium-ion battery soc estimation methods for electric vehicles," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2020, pp. 2385–2392.
- [48] V. Chandran, C. K Patil, A. Karthick, D. Ganeshaperumal, R. Rahim, and A. Ghosh, "State of charge estimation of lithium-ion battery for electric vehicles using machine learning algorithms," *World Electric Vehicle Journal*, vol. 12, no. 1, p. 38, 2021.
- [49] P. Venugopal and S. Reka S, "State of charge estimation of lithium batteries in electric vehicles using indrn," *IETE Journal of Research*, pp. 1–11, 2021.
- [50] J. P. Ortiz, J. D. Valladolid, C. L. Garcia, G. Novillo, and F. Berzeueta, "Analysis of machine learning techniques for the intelligent diagnosis of ni-mh battery cells," in *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. IEEE, 2018, pp. 1–6.
- [51] Z. Lou, J. Tuo, and Y. Wang, "Two-step principal component analysis for dynamic processes," in *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, May 2017, pp. 73–77.
- [52] K. Vatanparvar, S. Faezi, I. Burago, M. Levorato, and M. A. Al Faruque, "Extended range electric vehicle with driving behavior estimation in energy management," *IEEE transactions on Smart Grid*, vol. 10, no. 3, pp. 2959–2968, 2018.
- [53] G. Gruosso, G. Storti Gajani, F. Ruiz, J. D. Valladolid, and D. Patino, "A virtual sensor for electric vehicles' state of charge estimation," *Electronics*, vol. 9, no. 2, p. 278, 2020.
- [54] E. Redondo-Iglesias, P. Venet, and S. Pelissier, "Modelling lithium-ion battery ageing in electric vehicle applications—calendar and cycling ageing combination effects," *Batteries*, vol. 6, no. 1, p. 14, 2020.
- [55] X. Qin, M. Gao, Z. He, and Y. Liu, "State of charge estimation for lithium-ion batteries based on narx neural network and ukf," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, July 2019, pp. 1706–1711.
- [56] L. Wang, L. Wang, W. Liu, and Y. Zhang, "Research on fault diagnosis system of electric vehicle power battery based on obd technology," in *2017 International Conference on Circuits, Devices and Systems (ICCDs)*, Sep. 2017, pp. 95–99.
- [57] K. Khorsravinia, M. K. Hassan, R. Z. A. Rahman, and S. A. R. Al-Haddad, "Integrated obd-ii and mobile application for electric vehicle (ev) monitoring system," in *2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Oct 2017, pp. 202–206.
- [58] C. Sun, M. Pan, Y. Wang, J. Liu, H. Huang, and L. Sun, "Method for electric vehicle charging port recognition in complicated environment based on cnn," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 597–602.



Juan P. Ortiz obtained his B.Sc. in Electronic Engineering and his M.Sc. in Industrial Automation and Control from Universidad Politécnica Salesiana, in 2010 and 2014 respectively. He is professor of Automotive Engineering at the Universidad Politécnica Salesiana. He is a member of the Grupo de Investigación en Ingeniería del Transporte (GIIT). His research interests are nonlinear control theory, intelligent vehicles, electric vehicles and artificial intelligence.



Germán Ayabaca was born in Cuenca, Ecuador. He is a student of Automotive Mechanical Engineering from Universidad Politécnica Salesiana. He is currently a member of the Transportation Engineering Research Group (GIIT). His research interests are electrically powered vehicles and automotive technologies.



Ángel Cárdenas was born in Cuenca, Ecuador. He is a student of Automotive Mechanical Engineering from Universidad Politécnica Salesiana. He is currently a member of the Transportation Engineering Research Group (GIIT). His research interests are artificial neural networks and software engineering.



Diego Cabrera received his B.Sc. in Electronic Engineering from the Universidad Politécnica Salesiana, Ecuador, in 2012. He received his M.Sc. degree in Logic, Computation and Artificial Intelligence, and his Ph.D. degree in Computer Science from the Seville University, Spain, in 2014 and 2018 respectively. Currently, He is a Postdoctoral Fellow at the Dongguan University of Technology, China, and a Professor at the Universidad Politécnica Salesiana, Ecuador, from 2014. His research interests include intelligent systems, data-driven modelling

and fault diagnosis.



Juan D. Valladolid was born in Cuenca, Ecuador. He received the B.Sc. degree in electronic engineering from Universidad Politécnica Salesiana, Cuenca, Ecuador, the M.Sc. degree in Industrial Automation and Control from Universidad Politécnica Salesiana. He is currently pursuing the Ph.D degree, in engineering with the Pontificia Universidad Javeriana, Bogotá, Colombia. He is currently a Full-Time Professor with Universidad Politécnica Salesiana and member of the Grupo de Investigación en Ingeniería del Transporte (GIIT). He is the author of several articles. His current research interests include hybrid dynamical systems, power converter, systems optimization, electric vehicle, and nonlinear control theory.