REVISTA CUBANA
DE TRANSFORMACIÓN DIGITAL

RNPS 2487 • ISSN 2708-3411
Vol. 2 • Nro. 1 • enero-marzo 2021 • págs. 66-78

ARTÍCULO ORIGINAL

# Algoritmo de aprendizaje reforzado para software de modelado basado en Mapas Cognitivos Difusos

## *Reinforcement Learning Algorithm for Modeling Software based on Fuzzy Cognitive Maps*

*Ivan Santana*
*ching@uclv.edu.cu • http://orcid.org/0000-0001-5089-520X*
*Ariel Barreiros-Albo*
*abarreiros@uclv.cu • http://orcid.org/0000-0002-6238-069X*
*Richar Sosa*
*rslopez@uclv.edu.cu • http://orcid.org/0000-0003-3995-9895*
**UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS, CUBA**

*Recibido: 2020-10-31 • Aceptado: 2021-01-19*

**RESUMEN**

Los mapas cognitivos difusos son una herramienta potente con la que se puede llegar a modelar sistemas complejos con dinámicas indeterminadas, además de ser interpretables. Sin embargo, en ocasiones es difícil determinar con precisión las relaciones que se producen entre los conceptos de un sistema. En investigaciones previas se diseñó y desarrolló una biblioteca de *software* capaz de crear este tipo de modelos, y ajustarlos con buena precisión. Para lograr un buen ajuste de las matrices de pesos de un modelo que utilice el algoritmo de aprendizaje disponible, es necesario que se desarrolle a partir de un conjunto de valores específicos. En esta investigación se introdujo un nuevo algoritmo de aprendizaje automático a la biblioteca, que emplea técnicas de aprendizaje reforzado. Esto permite un mejor ajuste de las matrices de pesos, incluso al enfrentarse el aprendizaje a incertidumbre en la inicialización de los valores del modelo. Los resultados reflejan que un modelo que se obtiene mediante el empleo de la biblioteca con las modificaciones, se ajusta correctamente al comportamiento del sistema que emula en un mayor número de situaciones. La calidad del modelo se relaciona directamente con las iteraciones que se realicen para entrenarlo, siendo favorable un aumento de las mismas. Para la obtención de los resultados

se emplearon datos de simulación de un circuito RLC, a los cuales se le adicionó una señal de ruido para lograr una mayor semejanza a datos de procesos reales.

**Palabras clave:** aprendizaje automático; aprendizaje reforzado; mapas cognitivos difusos.

## ABSTRACT

*Fuzzy Cognitive Maps are a powerful tool that can be used to model complex systems with undetermined dynamics, in addition to being interpretable. However, sometimes it is difficult to determine precisely the relationships that occur between the concepts of a system. In previous research, a software library was designed and developed that is capable of creating this type of model and adjusting it with good precision. In order to achieve a good fit of the weight matrices of a model using the available learning algorithm, it is necessary to develop it from a specific set of values. In this research, a new Automatic Learning algorithm was introduced to the library, which uses Reinforced Learning techniques. This allows for better adjustment of the weight matrices, even when the learning is faced with uncertainty in the initialization of the model values. The results reflect that a model obtained using the library with the modifications, fits correctly to the behavior of the system that emulates in a greater number of situations. The quality of the model is directly related to the iterations that are made to train it, being favorable an increase of them. To obtain the results, simulation data of an RLC circuit was used, to which a noise signal was added to achieve a greater similarity to real process data.*

**Keywords:** *Machine Learning; Reinforcement Learning; Fuzzy Cognitive Maps.*

## INTRODUCCIÓN

La inteligencia artificial (IA), desde su surgimiento en el siglo XX, se ha ramificado considerablemente, mucha de las teorías que comprende han llegado a encontrar aplicación práctica en los sectores más vanguardistas del desarrollo científico-técnico (Topol, 2019; Venkatasubramanian, 2019; Yin, Li, Gao y Kaynak, 2014; Yousefi y Amoozandeh, 2016). La utilización de las diferentes técnicas del campo de la IA es costosa desde el punto de vista computacional, sin embargo, esta es una limitación que se ha tratado de solucionar con el paso de los años. En la actualidad, y con los potentes sistemas computacionales con los que se cuenta, es posible

modelar sistemas con altísima complejidad, que contemplan gran número de factores y las relaciones que se establecen entre ellos. También es posible el arribo a conclusiones a partir del análisis de cantidades de datos masivas (Cielen, Meysman y Ali, 2016; George, Osinga, Lavie y Scott, 2016).

Varias de las aplicaciones en varias ramas de investigación están basadas en modelos conexionista (Chen, 2018; Hirasawa, *et al.*, 2018; Mendonça, Chrun, Neves Jr. & Arruda, 2017). Este movimiento, en el contexto de la IA, intenta explicar las habilidades intelectuales de un sistema que utiliza pequeñas redes de unidades sencillas interconectadas, entre las que se encuentran las redes neuronales artificiales, los Mapas Cognitivos Difusos (MCD) y otras estructuras semejantes.

Un MCD es un gráfico que consiste en nodos y conexiones de fuerza entre estos nodos, que representan la fuerza de interacción entre los mismos. Cada nodo, para el MCD, representa un concepto que se utiliza para describir el comportamiento del sistema, y estos están conectados por arcos con signo y ponderados, que indican las relaciones causales que existen entre los conceptos (Kosko, 1986). La eliminación de deficiencias, como la estimación abstracta de la matriz de peso inicial y la dependencia del razonamiento subjetivo de los expertos, mejorará significativamente la funcionalidad de un MCD. En este contexto, el desarrollo de algoritmos de aprendizaje es un tema de investigación importante. Se han propuesto algunos algoritmos para el aprendizaje de MCD, en que la tarea principal del procedimiento es encontrar una configuración correcta de la matriz de pesos del modelo, que lo lleve al comportamiento estable deseado. Esto se logra a través de la minimización de una función objetivo adecuadamente definida. Los algoritmos establecidos dependen en gran medida de la aproximación inicial de la matriz de peso, que proporcionan los expertos (Jenitha & Kumaravel, 2014).

En investigaciones precedentes (Sosa, *et al.,* 2019) se desarrolló una biblioteca de *software* programada en *Python,* la cual permite la obtención de modelos basados en la teoría de mapas cognitivos difusos, haciendo pequeñas modificaciones a la misma. Como resultado se obtuvo un *software* capaz de crear modelos matemáticos de sistemas discretos. La biblioteca es capaz de modelar tanto sistemas que obedezcan a la teoría clásica de MCD, como sistemas con entradas por lo tanto posibilita tener en cuenta la interacción con variables externas. El algoritmo de aprendizaje utilizado para la creación de la biblioteca de modelado en cuestión, emplea el método del gradiente descendente para minimizar una función determinada. La función a minimizar es el error cuadrático medio, que representa el error entre los valores de un *dataset* determinado y las predicciones que realiza el algoritmo para las mismas condiciones (Sosa, *et al.,* 2019).

El objetivo fundamental de este trabajo es diseñar e implementar un nuevo algoritmo de aprendizaje que permita disminuir las limitaciones del método del gradiente descendente. Para ello se propone emplear aprendizaje reforzado (AR), lo que posibilitará la creación de modelos de sistemas en un mayor número de situaciones. Se evaluará experimentalmente el algoritmo creado, analizando las potencialidades que este ofrece en distintas situaciones.

## METODOLOGÍA

De acuerdo con lo planteado en la teoría de MCD, pudiera llegarse a la ecuación de inferencia del mapa [ecuación (1)], que introduce un par de nuevos parámetros a la teoría clásica, para dotar a los modelos de la capacidad de modelar procesos con entradas (Sosa, *et al.,* 2019). En este caso la fuerza de dicha interacción entre los conceptos está definida por una matriz de pesos *W,* mientras que $U^k$ representa a las entradas independientes que presenta el modelo, y *H* representa el acoplamiento entre estas entradas y los conceptos, y define la fuerza de las interacciones que se producen.

$$A^{(k+1)} = f\left(WA^k + HU^k\right) \quad (1)$$

La función *f* empleada para la acotación del mapa es una función por tramos, utilizada en Sosa, *et al.,* (2019) [ecuación (2)], y se satura en límites definidos por los expertos.

$$f(x) = \begin{cases} min, \forall\, x < \text{mín} \\ x, \forall\, min \leq x \leq \text{max} \\ max, \forall\, x > \text{máx} \end{cases} \quad (2)$$

Al tratarse el modelado de un proceso multi-variable, se establecen diversas relaciones de dependencia entre los conceptos que comprende el modelo, las cuales son difíciles de definir con exactitud, incluso por personas expertas en el proceso que se pretende modelar. La dificultad estriba en determinar los valores de las matrices *W* y *H* del modelo de un sistema, ya que estas establecen los valores de dependencia entre los conceptos y las entradas. Estas matrices son llamadas matrices de peso del modelo, y del buen ajuste de las mismas dependerá en gran medida el grado de exactitud de las predicciones del modelo, la obtención del comportamiento deseado, y por lo tanto la calidad del mismo.

Si se cuenta con un conjunto de datos para el proceso de aprendizaje, y se considera $x_{nDs}^{k+1}$ el valor esperado para el n-ésimo concepto en el instante *k*+1 según el *dataset,* y a $x_n^{k+1}$ como el valor de inferencia del mapa en el mismo instante de tiempo. Se podrá entonces definir el error cuadrático medio según se muestra en la ecuación (3).

$$E_n^{k+1} = \frac{1}{2}\left(x_n^{k+1} - x_{nDs}^{k+1}\right)^2 \quad (3)$$

Al sustituir la ecuación de inferencia del mapa (1) en la ecuación (3) se obtiene:

$$E_n^{k+1} = \frac{1}{2}\left(f\left(WA^k + HU^k\right) - x_{nDs}^{k+1}\right)^2 \quad (4)$$

Al derivar esta función de error para cada uno de los parámetros de la matriz *W* se obtiene la ecuación (5) que describe el comportamiento del error cuadrático medio para el término n.

Un procedimiento similar se utiliza para obtener la función del error para los valores de activación de las entradas.

$$\frac{\partial E_n^{k+1}}{\partial w_{an}} = \left(f\left(WA^k + HU^k\right) - x_{nDs}^{k+1}\right) f^{\cdot}\left(WA^k + HU^k\right) x_a^k \quad (5)$$

Dada la función del error a minimizar para cada concepto, es difícil conocer a priori el comportamiento que va a tener, por lo que es posible que tenga más de un mínimo. El método del gradiente descendente garantiza que la función para una cantidad de iteraciones suficientes alcanza un mínimo, sin embargo, no hay forma de saber o garantizar que este es el mínimo global de la función, o en todo caso, si existe un valor mínimo para esta función.

El empleo de un nuevo algoritmo es necesario con el objetivo de lograr la convergencia a los valores óptimos de los parámetros del modelo. Para determinar los factores que afectan el desempeño del aprendizaje se analizó el mismo bajo diferentes circunstancias. En la biblioteca del software que se tomó como punto de partida el entrenamiento del modelo se produce a partir de un punto determinado por una matriz de pesos aleatoria. Dicha matriz de pesos se genera a partir de una máscara, que determina tanto el signo como el valor máximo que pueden tomar cada causalidad dentro de *W* y *H,* antes de comenzar el proceso.

Al evaluar el comportamiento del aprendizaje bajo varias circunstancias se puede observar que uno de los factores que más afecta el resultado del proceso de aprendizaje es la selección de un punto de partida adecuado para comenzar el entrenamiento del modelo. Este punto de partida se determina a partir de las máscaras de las causalidades planteadas anteriormente, pero también depende del valor máximo que se determine por parte de los expertos. Si las causalidades con las que se inicializa el modelo están muy alejadas de un rango acotado de valores definidos por expertos y determinados por el proceso, es muy probable que el resultado obtenido luego del aprendizaje sea inaceptable, y las predicciones erradas.

La utilización de AR en este caso, para solucionar las dificultades que se presentan en el aprendizaje, contrasta con otros enfoques de aprendizaje automático que se pudieran aplicar, ya que en este al algoritmo no se le dice explícitamente cómo realizar una tarea, sino que resuelve el problema por sí solo (Fang, Li & Cohn, 2017). Otro aspecto importante de AR es que utiliza la experiencia de prueba y error (a diferencia de otros métodos, que asumen un conocimiento completo del entorno a priori). Por lo tanto, el algoritmo de AR no requiere un conocimiento completo o control del entorno, solo necesita interactuar con este y recopilar información. En un ambiente fuera de línea, primero se adquiere la experiencia y luego se utilizan los resultados obtenidos (François-Lavet, *et al.,* 2018; Polydoros & Nalpantidis, 2017).

En los casos en los que se cuente con una amplia información sobre el proceso, el punto de partida adecuado puede ser determinado con un buen nivel de precisión. Existen, sin embargo, un gran número de procesos en los que las dinámicas internas y las dependencias entre los conceptos y entradas del proceso no están claramente definidas, incluso por expertos en el mismo. Otro factor que influye en la obtención de un buen modelo es la cantidad de datos

disponibles durante el entrenamiento (Lange, Gabel & Riedmiller, 2012). Cuando se cuenta con gran cantidad de datos el entrenamiento tiende a ser más tolerante, y se suele llegar a resultados más precisos (Wu, *et al.,* 2018; Zhang, Han & Deng, 2018).

La potenciación al *software* realizada en esta investigación viene dada por la inclusión de un nuevo método de aprendizaje que emplea AR. Esto permite que se mejore también la capacidad de generalización del *software.* Se parte del precepto de que el sistema no conocerá cuáles acciones deberá tomar, en cambio deberá descubrir por sí mismo cuáles ofrecen una mayor recompensa en cada situación o estado, mediante prueba y error. Para la implementación del aprendizaje mediante AR se tomó la política de selección *ε-greedy,* que tiene a *ε* como único parámetro, y este representa la probabilidad de selección de una acción que no tiene la mayor recompensa (probabilidad de exploración) (Sewak, 2019). Existen numerosas políticas de selección que mantienen la relación exploración-explotación balanceada, cada una de ellas tiene características específicas, pero el objetivo general a través de las mismas se mantiene. La selección se debe a la sencilla implementación y compresión de dicha política, y a los buenos resultados que presenta en casos prácticos (Sutton & Barto, 2018).

El nuevo algoritmo que se introduce deberá encontrar una combinación de valores a partir de los cuales comenzará a ejecutarse el método del gradiente descendente. Una vez tomadas las acciones correspondientes en cada estado se ejecuta el método del gradiente descendente con un número reducido de iteraciones. El objetivo en este caso no es obtener los parámetros del modelo, sino conocer mediante el cálculo del error cuadrático medio, cuán bueno es el conjunto de valores seleccionados.

Cuando el algoritmo determina que un conjunto de valores ha alcanzado un error menor que el que se tenía constancia hasta ese momento, entonces se recompensa a este conjunto de valores, propiciando su reelección. La ecuación (6) representa la función que se emplea para generar una recompensa *R* para dicho conjunto, donde *e* es el error cuadrático medio al que arriba al algoritmo de evaluación. Esta tiene un comportamiento adecuado para esta aplicación, ya que para un menor valor de error entrega una mayor recompensa, y se alcanzaría el valor cero solo con un error infinito.

$$R = (0.98^{e}) \times 100 \quad (6)$$

Durante el proceso de creación y ajuste de un modelo creado al utilizar este software, existen dos formas diferentes de alcanzar el estado en que el aprendizaje ha finalizado. La forma ideal sería garantizar que el error que se obtiene al comprobar el desempeño del modelo obtenido sea menor que un error mínimo aceptable, sin embargo, en muchas ocasiones este mínimo nunca llega a alcanzarse pues es demasiado pretencioso. La segunda forma de finalizar el aprendizaje es mediante la planificación de un número de iteraciones. Para ello debe tenerse en cuenta que un número demasiado alto puede comprometer la capacidad de generalización del modelo, al realizarse un sobreajuste de los parámetros. Por otra parte, un número demasiado bajo puede significar que el modelo nunca llegue a alcanzar el comportamiento deseado,

por ser interrumpido el proceso antes de finalizar. Debe establecerse por lo tanto una relación de compromiso, en este caso un buen valor de este parámetro puede ser alcanzado mediante experimentación.

Dada las características del método de aprendizaje, si se toma i como el número de iteraciones que realiza el algoritmo, cuando $i \to \infty$ debe alcanzarse un conjunto de valores a partir del cual al ejecutar el método del gradiente descendente se arribe a un modelo con un comportamiento adecuado. Partiendo del precepto de que al finalizar las iteraciones planificadas se tendrá un conjunto de valores de inicialización adecuados para las matrices de pesos *W* y *H,* se ejecuta el método del gradiente descendente, heredado de Sosa, *et al.,* (2019), y de esta forma finaliza el aprendizaje.

El algoritmo desarrollado mantiene compatibilidad con los modelos creados anteriormente y puede emplearse de forma escalonada o independiente, para obtener modelos de plantas. Sin embargo, se introduce en esta investigación una manera alternativa para el ajuste de las matrices de peso de los modelos, en lugar de emplear el gradiente descendente como única técnica. De esta manera el proceso de ajuste sináptico se desarrolla como se muestra en la figura 1.

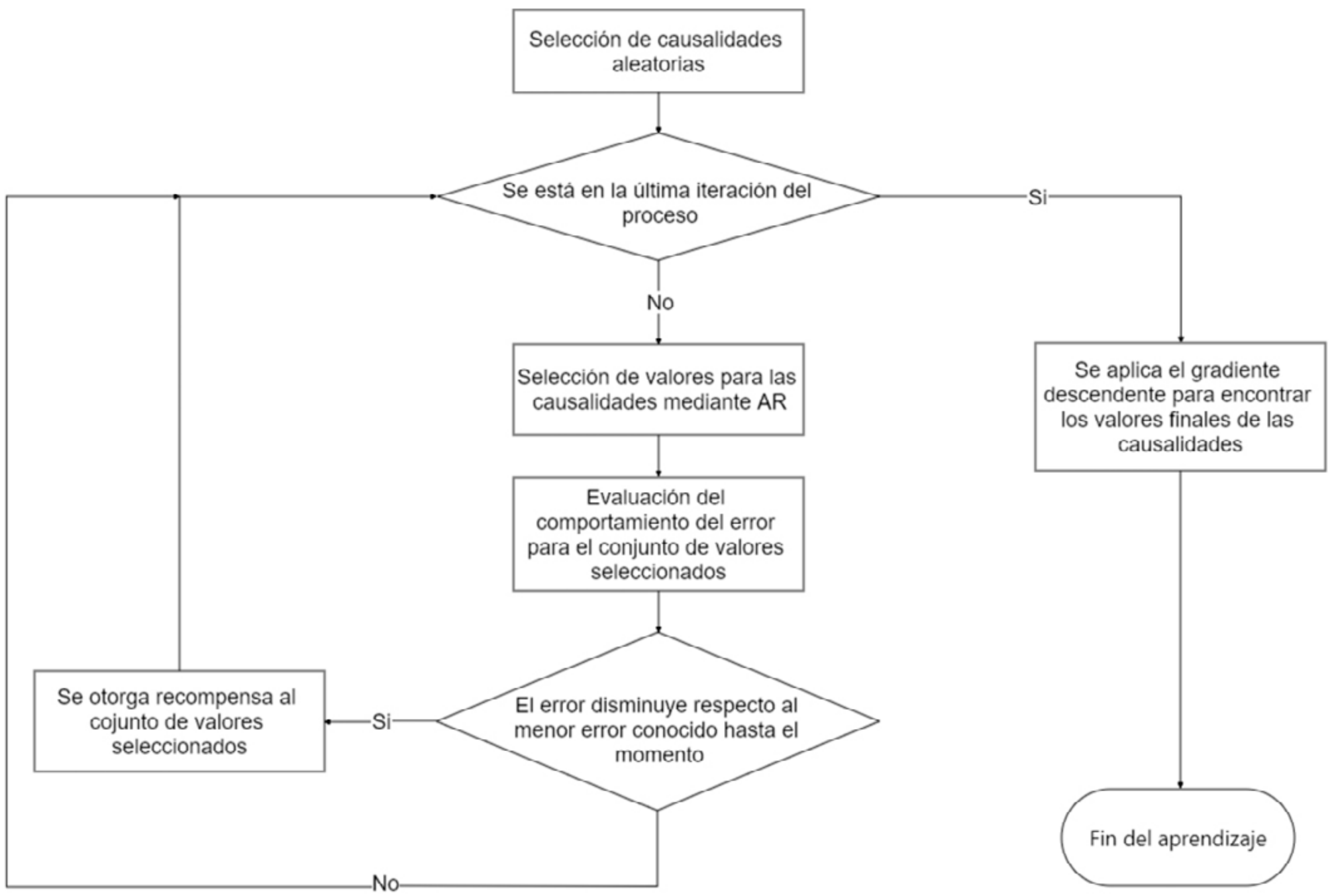


**Figura 1. Flujo del proceso de creación de un modelo empleando AR.**

## Análisis de los resultados experimentales

Para comprobar el desempeño del *software* desarrollado en esta investigación se realizó un experimento en el que se comparan los resultados obtenidos, se emplearon las técnicas de aprendizaje

desarrolladas en esta investigación, con respecto a datos de simulación a los que se le ha introducido ruido. El algoritmo de aprendizaje se enfrentó a la posibilidad de que las causalidades con las que se inicializa el modelo se generen en un intervalo más amplio, condiciones estas en la que el aprendizaje, tal como se realizaba anteriormente (Sosa, *et al.,* 2019), presentaba dificultades. El objetivo de esta prueba fue comprobar que el nuevo algoritmo de aprendizaje tiene una mayor capacidad de generalización, que le permita enfrentarse de una mejor manera a situaciones en las que se desconozcan o se conozcan con poca precisión ciertas características de un proceso.

Para todo ello se empleó el circuito RLC que se muestra en la figura 2, y cuyas dinámicas son conocidas, por lo que al final se pudo determinar con mayor certeza la calidad del modelo obtenido. Las variables que se utilizaron para creación y validación del modelo son los voltajes en los capacitores $C_1$ y $C_2$ ($Vc_1$ y $Vc_2$ respectivamente) y la corriente IL a través del inductor, se tomó como entrada del modelo al voltaje $V_i$, coincidentemente con la fuente de alimentación del circuito. Se procuró que el modelo matemático del sistema empleado fuera un sistema de tercer orden, con el objetivo de demostrar las capacidades del *software* para modelar sistemas descritos por ecuaciones de orden superior.

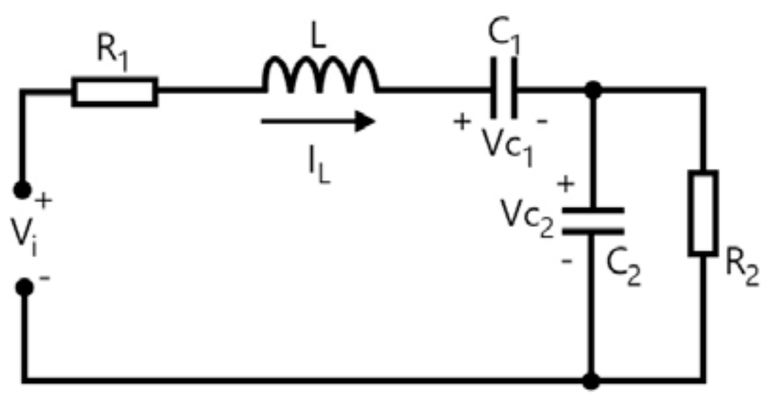


**Figura 2. Circuito RLC empleado para las pruebas.**

Durante la prueba realizada se evaluó el desempeño de los modelos obtenidos al finalizar el aprendizaje, al generarse las matrices de causalidad en un rango más amplio. En este caso se partió de acotar un intervalo de inicialización en el que se conocía que el aprendizaje lograba buenos resultados con el algoritmo anterior, que no empleaba AR durante el aprendizaje, este intervalo se amplió en un 300 % para realizar la prueba. De la totalidad de los datos disponibles del proceso se empleó el 70 % para llevar a cabo el entrenamiento del modelo, y el 30 % restante para validar los resultados.

En la figura 3 se observa el comportamiento de las predicciones de los modelos a los que se arribó utilizando el nuevo algoritmo desarrollado y el algoritmo previamente existente, así como el comportamiento del error cuadrático medio en las predicciones, tomando para la inicialización de los valores el intervalo ampliado. En esta prueba el algoritmo que no emplea AR no logró un buen ajuste de las matrices de peso al enfrentarse a una mayor incertidumbre en la inicialización de los valores. Por otra parte, el nuevo algoritmo desarrollado se comporta de forma adecuada, que las predicciones que se realizan con los modelos obtenidos mantuvieron el error cuadrático medio en valores bajos en todo momento.

La figura 4 muestra el desempeño del nuevo algoritmo al desarrollarse el aprendizaje partiendo intervalos de inicialización tanto acotados como ampliados. El desempeño que se ob-

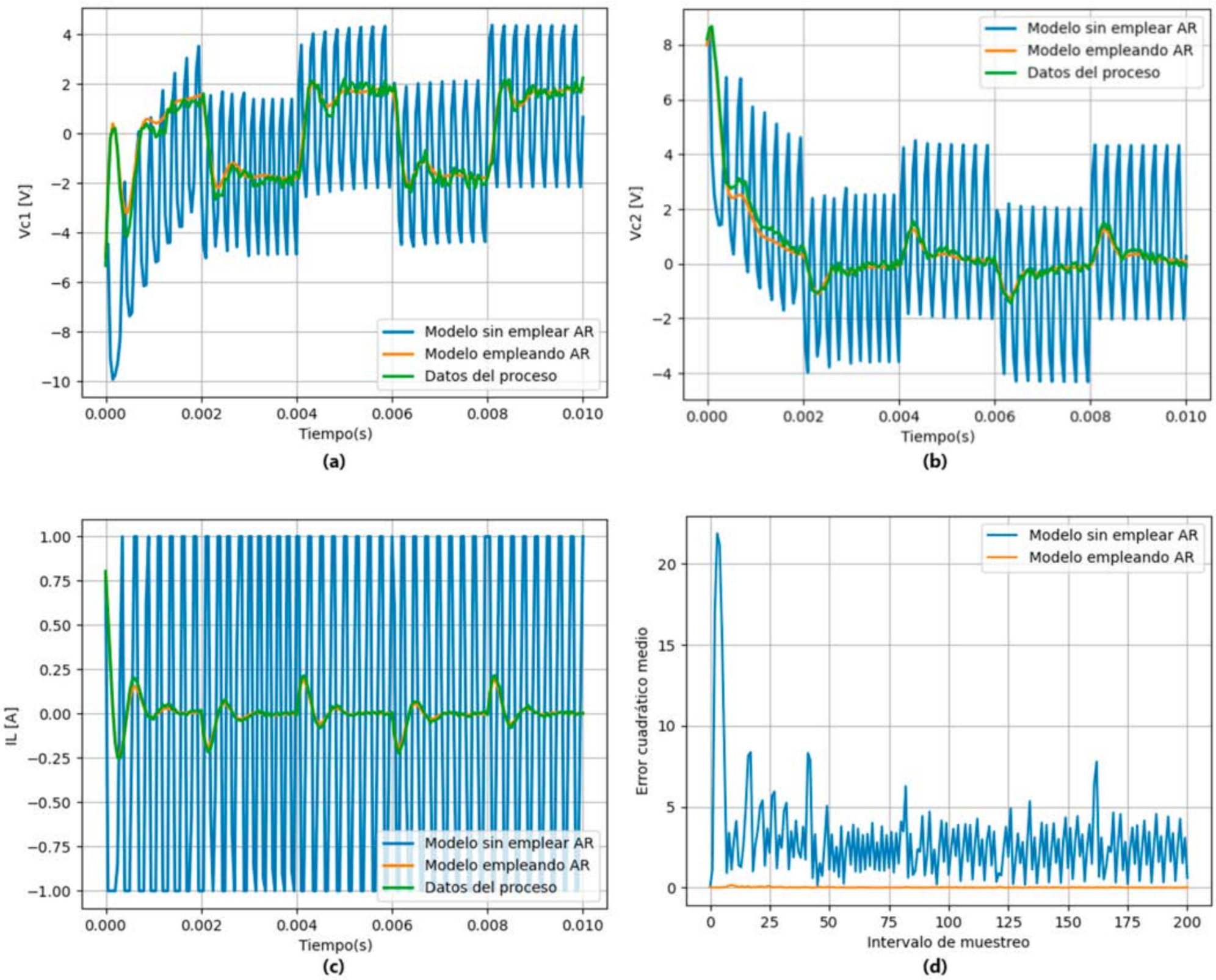


**Figura 3. (a) Comportamiento de modelos con y sin emplear AR para el parámetro Vc1. (b) Comportamiento de modelos con y sin emplear AR para el parámetro Vc2. (c) Comportamiento de modelos con y sin emplear AR para el parámetro IL. (d) Comportamiento del error cuadrático medio de los modelos (a), (b) y (c).**

tiene es en ambos casos similar al obtenido en Sosa, *et al.,* (2019), manteniéndose el error cuadrático medio en valores bajos durante las predicciones, y denotando la mayor capacidad de generalización del nuevo algoritmo.

La figura 5 muestra la evolución del error cuadrático medio a través del aprendizaje, en esta se puede apreciar cómo el error va disminuyendo paulatinamente a medida que el algoritmo va explorando nuevas posibilidades. Puede apreciarse el balance entre explotación y exploración que aporta el empleo de AR al analizar la forma en la que se desarrolla el ajuste de los pesos sinápticos de los modelos creados.

En la ampliación del comportamiento que se muestra en la figura 5, los puntos 1 y 3 indican que, durante el aprendizaje, el algoritmo exploró nuevos valores que redujeron el valor del error, los cuales fueron recompensados. Mientras tanto en los puntos 2 y 6 la exploración condujo a valores que, contrariamente a lo deseado, aumentaron el error del modelo. Por otra parte, en el punto 5 el algoritmo no explora nuevos valores, sino que explota los que previamente se habían recompensado como buena combinación en el punto 4. Esta tendencia del aprendizaje a explorar todas las posibilidades provoca que las posibilidades de encontrar un

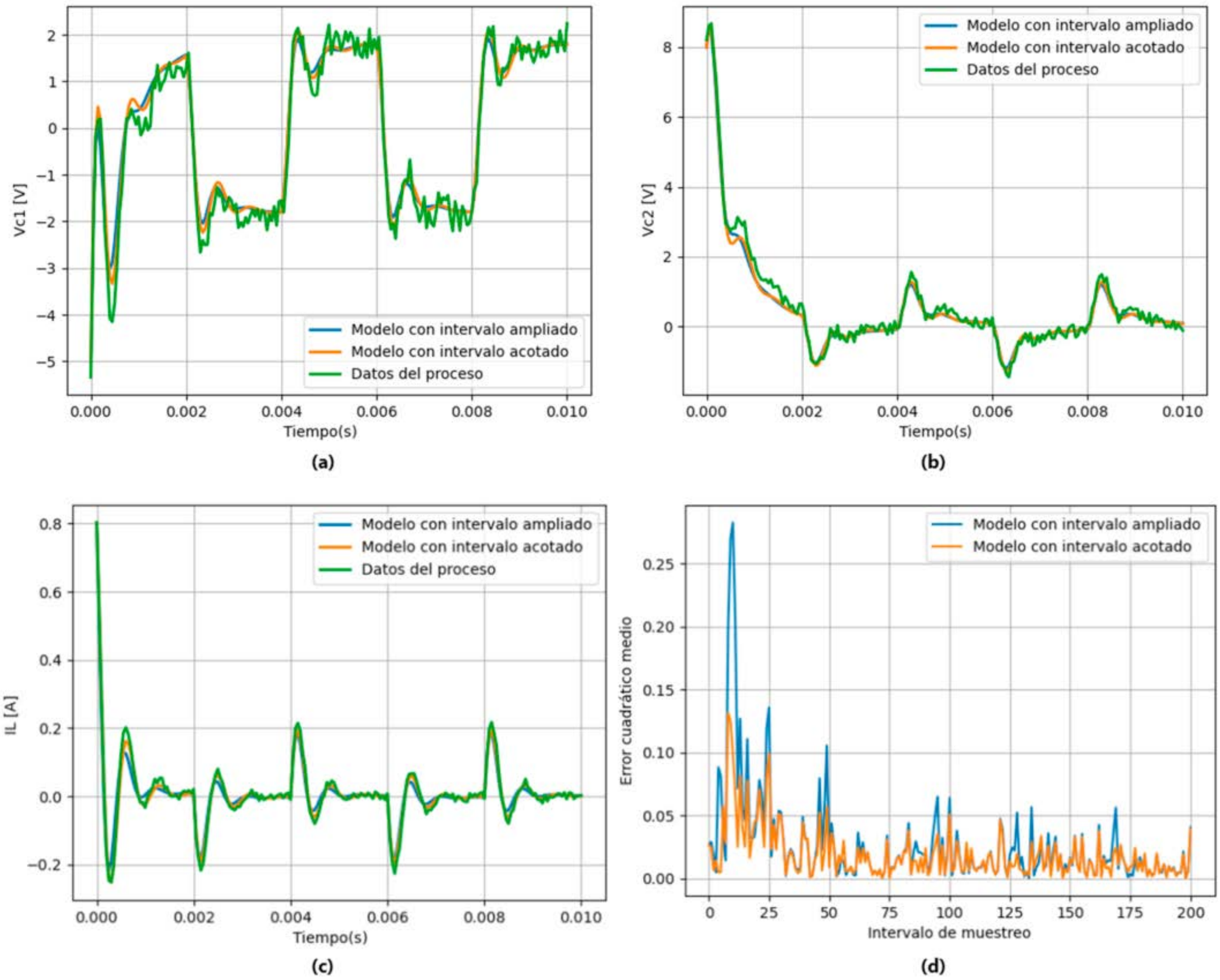


**Figura 4. (a) Comportamiento de modelos empleando AR para el parámetro Vc1. (b) Comportamiento de modelos empleando AR para el parámetro Vc2. (c) Comportamiento de modelos empleando AR para el parámetro IL. (d) Comportamiento del error cuadrático medio de los modelos de (a), (b) y (c).**

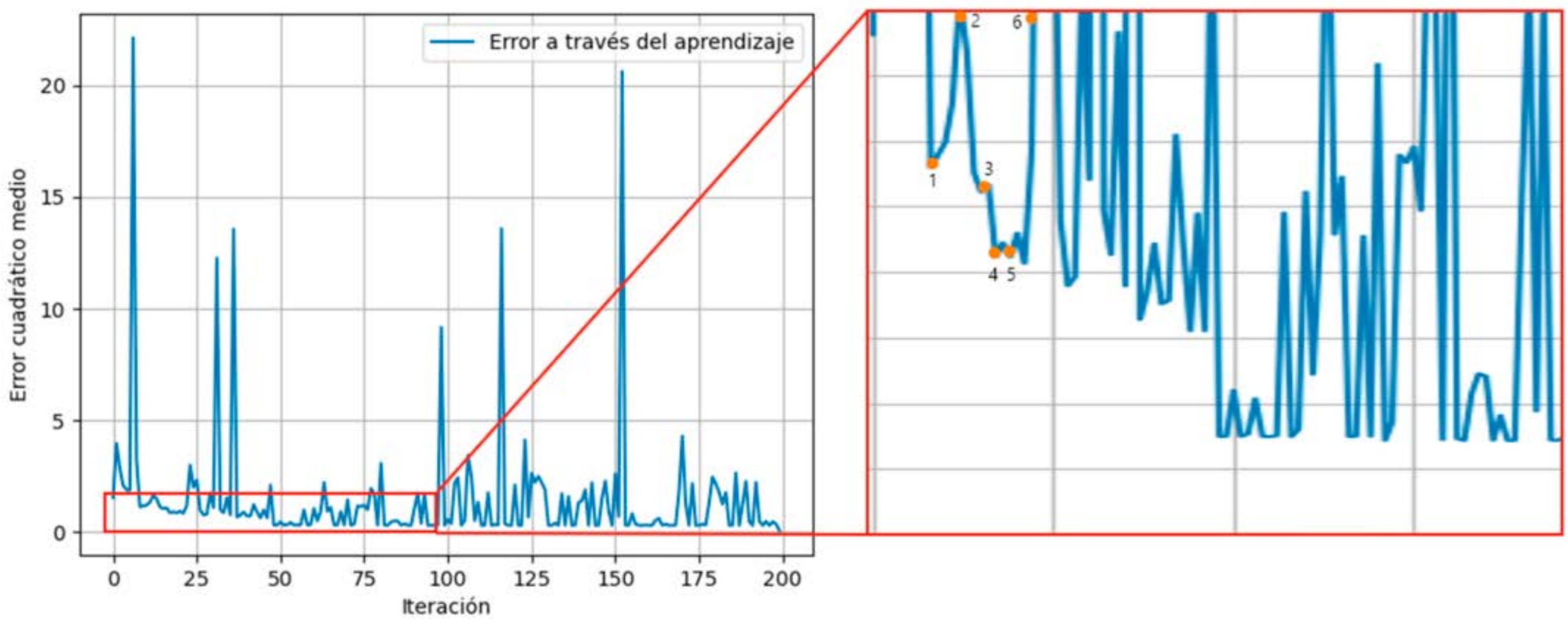


**Figura 5. Evolución del aprendizaje con técnicas de AR.**

buen ajuste se incrementen con el incremento de las iteraciones que se planifican para realizar el aprendizaje. Lograr un buen balance entre la explotación y la exploración en el proceso de

aprendizaje, junto a la selección de una cantidad adecuada de pasos a realizar por el aprendizaje, logra a menudo que este se realice de forma satisfactoria. Lograr un balance entre estos factores depende de cada aplicación en particular y del resultado que se desee, y actualmente representa un desafío para los expertos. Al incrementar demasiado, por ejemplo, el número de iteraciones a realizarse, aumenta también el costo computacional del algoritmo, y puede llegar a producir un sobreajuste del modelo, cuestión no deseada.

## CONCLUSIONES

En esta investigación se incorporó a la biblioteca de *software* para modelado de sistemas discretos mediante mapas cognitivos difusos, un nuevo método que emplea aprendizaje reforzado para mejorar los modelos que se obtienen. Se empleó un sistema cuyas dinámicas son conocidas, con el objetivo de poder evaluar los resultados, pero se procuró que dicho sistema tuviera dinámicas complejas que se asemejaran a las que se pudieran presentar en procesos reales. A partir de los resultados obtenidos experimentalmente, se observó que el nuevo algoritmo introducido disminuye el error continuamente a medida que se incrementa el número de iteraciones durante el aprendizaje. También se apreció que este algoritmo de aprendizaje es menos vulnerable a una selección no acotada en la inicialización de las matrices de peso, aspecto que afecta negativamente la calidad del modelo. Este problema puede atenuarse al considerar el criterio de expertos, los cuales definen intervalos adecuados para inicializar las matrices sobre la base de su conocimiento del proceso.

Los resultados obtenidos pueden ser empleados en procesos cuyas dinámicas que definen su comportamiento no están bien determinadas existiendo relaciones difusas entre ellas. La robustez y capacidad de generalización del algoritmo propuesto, permite su utilización en situaciones donde existe poco conocimiento de las relaciones causales entre variables, o incluso a donde no exista conocimiento experto alguno para definir un modelo inicial a partir del cual parta el modelado.

El uso del algoritmo de aprendizaje propuesto, posibilitaría resultados superiores en investigaciones previas. En la predicción de parámetros de crecimiento y desarrollo de hortalizas (Madruga, *et al.,* 2019) hubo una afectación en el modelado por la limitación en la cantidad de datos adquiridos, así como incertidumbre en algunas relaciones entre variables y valores de los pesos iniciales. Estos aspectos pueden ser resueltos a partir de la utilización del algoritmo de aprendizaje utilizando AR.

## REFERENCIAS

Chen, R. Y. (2018). A traceability chain algorithm for artificial neural networks using T–S fuzzy cognitive maps in blockchain. *Future Generation Computer Systems, 80,* 198-210.

Cielen, D., Meysman, A. & Ali, M. (2016). *Introducing data science: big data, machine learning, and more, using Python tools:* Manning Publications Co.

Fang, M., Li, Y. & Cohn, T. (2017). *Learning how to Active Learn: A Deep Reinforcement Learning Approach.* Paper presented at the Conference on Empirical Methods in Natural Language Processing.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, MG. & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations Trends® in Machine Learning, 11*(3-4), 219-354.

George, G., Osinga, E. C., Lavie, D. & Scott, B. A. (2016). Big data and data science methods for management research. In: Academy of Management Briarcliff Manor, New York.

Hirasawa, T., Aoyama, K., Tanimoto, T., Ishihara, S., Shichijo, S., Ozawa, T., . . . Fujisaki, J. (2018). Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images. *Gastric Cancer, 21*(4), 653-60.

Jenitha, G. y Kumaravel, A. (2014). An Instance of Reinforcement Learning Based on Fuzzy Cognitive Maps. *International Journal of Applied Engineering Research, 9*(18), 3913-20.

Kosko, B. (1986). Fuzzy cognitive maps. *International journal of man-machine studies, 24*(1), 65-75.

Lange, S., Gabel, T. & Riedmiller, M. (2012). Batch Reinforcement Learning. In M. Wiering y M. van Otterlo (Eds.), *Reinforcement Learning: State-of-the-Art* (pp. 45-73). Berlin, Heidelberg: Springer Berlin Heidelberg.

Madruga, A., Alvarado, Y., Sosa, R., Santana, I. y Mesa, J. R. (2019). Modelo de crecimiento y desarrollo de hortalizas en casas de cultivo mediante mapas cognitivos difusos. *Revista Cubana de Ciencias Informáticas, 13*(2), 47-60.

Mendonça, M., Chrun, I. R., Neves Jr, F. & Arruda, L. V. (2017). A cooperative architecture for swarm robotic based on dynamic fuzzy cognitive maps. *Engineering Applications of Artificial Intelligence, 59,* 122-132.

Polydoros, A. S. & Nalpantidis, L. (2017). Survey of model-based reinforcement learning: Applications on robotics. *Journal of Intelligent Robotic Systems, 86*(2), 153-173.

Sewak, M. (2019). Q-Learning in Code. In *Deep Reinforcement Learning* (pp. 65-74): Springer.

Sosa, R., Alfonso, A., Nápoles, G., Bello, R., Vanhoof, K. & Nowé, A. (2019). *Synaptic Learning of Long-Term Cognitive Networks with Inputs.* Paper presented at the 2019 International Joint Conference on Neural Networks (IJCNN).

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction:* MIT press.

Topol, EJ. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine, 25*(1), 44-56.

Venkatasubramanian, V. (2019). The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal, 65*(2), 466-478. doi:10.1002/aic.16489

Wu, L., Tian, F., Qin, T., Lai, J. & Liu, T.Y. (2018). A study of reinforcement learning for neural machine translation. *arXiv preprint arXiv:.08866.*

Yin, S., Li, X., Gao, H. & Kaynak, O. (2014). Data-based techniques focused on modern industry: An overview. *IEEE Transactions on Industrial Electronics, 62*(1), 657-67.

Yousefi, F. & Amoozandeh, Z. (2016). Statistical mechanics and artificial intelligence to model

the thermodynamic properties of pure and mixture of ionic liquids. *Chinese Journal of Chemical Engineering, 24*(12), 1761-71.

Zhang, D., Han, X. & Deng, C. (2018). Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE Journal of Power Energy Systems, 4*(3), 362-70.