

## Algoritmo de predicción con Redes Neuronales Recurrentes LSTM para crisis epiléptica basado en un brazalete inteligente

Prediction algorithm with LSTM Recurrent Neural Networks for epileptic seizures based on a smart bracelet

Algoritmo de previsão com redes neurais recorrentes LSTM para crises epilépticas com base numa pulseira inteligente

Anguaya-Otavallo, Nancy Zulay

Universidad Nacional de Chimborazo

[nancy.anguaya@unach.edu.ec](mailto:nancy.anguaya@unach.edu.ec)

<https://orcid.org/0009-0000-6175-8454>



Molina-Valdiviezo, Lorena

Universidad Nacional de Chimborazo

[lmolina@unach.edu.ec](mailto:lmolina@unach.edu.ec)

<https://orcid.org/0000-0001-9056-8262>



DOI / URL: <https://doi.org/10.55813/gaea/ccri/v6/n2/1232>

### Como citar:

Anguaya-Otavallo, N. Z., & Molina-Valdiviezo, L. (2025). Algoritmo de predicción con Redes Neuronales Recurrentes LSTM para crisis epiléptica basado en un brazalete inteligente. *Código Científico Revista De Investigación*, 6(2), 762–783.

Recibido: 23/11/2025

Aceptado: 24/12/2025

Publicado: 31/12/2025

## Resumen

Debido a la naturaleza impredecible de estas crisis neurológicas, si se detectan tardíamente las crisis epilépticas en ambientes no controlados, supone un peligro significativo para la integridad física de los pacientes. Los sistemas de monitoreo ambulatorio actuales, a pesar de los progresos realizados, no poseen la precisión requerida para diferenciar patrones complejos electroencefalográficos en tiempo real. Este análisis buscó determinar si una Red Neuronal Recurrente (RNN) de arquitectura Long Short-Term Memory (LSTM) tenía la capacidad de anticipar crisis epilépticas basándose en señales electroencefalográficas (EEG), con el propósito de respaldar sistemas de vigilancia ambulatoria y alerta temprana. La metodología empleada fue un enfoque de aprendizaje profundo aplicado a series temporales, a partir de un conjunto de datos públicos disponibles en la plataforma Kaggle para identificar crisis epilépticas. Se entrenó y validó un modelo de clasificación binaria usando 11,500 segmentos de señales EEG procesados. El preprocesamiento abarcó la división de las señales y la disposición de los datos en secuencias temporales que fueran apropiadas para la arquitectura de la red neuronal recurrente LSTM. Las métricas de clasificación estándar se utilizaron para evaluar el rendimiento del modelo. Los resultados indicaron que el modelo logró una precisión del 92,27 %, lo que demuestra su gran habilidad para diferenciar entre estados con crisis epiléptica y aquellos sin ella. Estos resultados corroboraron la hipótesis de que las redes LSTM tienen la capacidad de modelar patrones temporales complejos hallados en las señales EEG. El modelo sugerido resultó ser una herramienta computacional sólida para anticipar crisis epilépticas. La principal aportación del estudio fue mostrar que es posible incorporar modelos LSTM en dispositivos portátiles, como los brazaletes inteligentes. Esto genera nuevas posibilidades para crear sistemas de alerta temprana y enfoques de intervención clínica individualizada.

**Palabras clave:** crisis epiléptica, función de activación, función de perdida, long short-term memory, red neuronal recurrente.

## Abstract

Due to the unpredictable nature of these neurological events, delayed detection of epileptic seizures in uncontrolled environments poses a significant risk to patients' physical safety. Current ambulatory monitoring systems, despite advancements, lack the accuracy required to differentiate complex electroencephalographic patterns in real time. This analysis aimed to determine whether a Recurrent Neural Network (RNN) with a Long Short-Term Memory (LSTM) architecture could anticipate epileptic seizures based on electroencephalographic (EEG) signals, with the goal of supporting ambulatory monitoring and early warning systems. The methodology employed was a deep learning approach applied to time series analysis, using a publicly available dataset on the Kaggle platform for identifying epileptic seizures. A binary classification model was trained and validated using 11,500 processed EEG signal segments. The preprocessing involved splitting the signals and arranging the data into temporal sequences appropriate for the LSTM recurrent neural network architecture. Standard classification metrics were used to evaluate the model's performance. The results indicated that the model achieved 92.27% accuracy, demonstrating its strong ability to differentiate between seizure states and non-seizure states. These results corroborated the hypothesis that LSTM networks can model complex temporal patterns found in EEG signals. The proposed model proved to be a robust computational tool for anticipating seizures. The main contribution of the study was to show that it is possible to incorporate LSTM models into wearable devices, such as smart bracelets. This opens up new possibilities for creating early warning systems and individualized clinical intervention approaches.

**Keywords:** epileptic seizure, activation function, loss function, long short-term memory, recurrent neural network.

## Resumo

Devido à natureza imprevisível desses eventos neurológicos, a deteção tardia de crises epilépticas em ambientes não controlados representa um risco significativo para a segurança física dos pacientes. Os atuais sistemas de monitorização ambulatória, apesar dos avanços, carecem da precisão necessária para diferenciar padrões eletroencefalográficos complexos em tempo real. Esta análise teve como objetivo determinar se uma Rede Neural Recorrente (RNN) com uma arquitetura de Memória de Curto Prazo Longo (LSTM) poderia antecipar crises epilépticas com base em sinais eletroencefalográficos (EEG), com o objetivo de apoiar a monitorização ambulatória e os sistemas de alerta precoce. A metodologia empregada foi uma abordagem de aprendizagem profunda aplicada à análise de séries temporais, utilizando um conjunto de dados disponível publicamente na plataforma Kaggle para identificar crises epilépticas. Um modelo de classificação binária foi treinado e validado utilizando 11.500 segmentos de sinais EEG processados. O pré-processamento envolveu a divisão dos sinais e a organização dos dados em sequências temporais apropriadas para a arquitetura da rede neural recorrente LSTM. Foram utilizadas métricas de classificação padrão para avaliar o desempenho do modelo. Os resultados indicaram que o modelo alcançou 92,27% de precisão, demonstrando a sua forte capacidade de diferenciar entre estados de convulsão e estados sem convulsão. Esses resultados corroboraram a hipótese de que as redes LSTM podem modelar padrões temporais complexos encontrados nos sinais de EEG. O modelo proposto provou ser uma ferramenta computacional robusta para antecipar convulsões. A principal contribuição do estudo foi mostrar que é possível incorporar modelos LSTM em dispositivos vestíveis, como pulseiras inteligentes. Isso abre novas possibilidades para a criação de sistemas de alerta precoce e abordagens de intervenção clínica individualizadas.

**Palavras-chave:** crises epilépticas, função de ativação, função de perda, memória de longo prazo, rede neural recorrente.

## Introducción

La red neuronal artificial (RNA) es un algoritmo de aprendizaje automático que tiene como modelo el cerebro humano. Su propósito es procesar datos de la misma forma en que lo hace este órgano. La RNA es una herramienta matemática poderosa y flexible que tiene la capacidad de manejar muchas tareas complicadas, como la aproximación de funciones, las series temporales y los problemas de clasificación. La habilidad más importante de una RNA es optimizar su desempeño mediante el aprendizaje basado en experiencias previas. La RNA tiene la capacidad de trabajar con datos ruidosos y aproximar funciones. Su naturaleza no lineal la hace altamente efectiva cuando se trata de resolver problemas en los que los métodos

convencionales no son efectivos, y es compatible con varios algoritmos de entrenamiento (Aggarwal, 2023).

Una RNA está formada por unidades conectadas entre sí, conocidas como neuronas, que se organizan en capas. Cada neurona, por medio de una función de activación, transforma las entradas que combina linealmente al sumar los productos de sus sesgos y pesos. La RNA produce una salida al optimizar la minimización de errores a través de un flujo de datos en progreso. Durante cada época de entrenamiento, actualiza los parámetros de la red mediante la retropropagación de errores. Hoy en día, las RNA se emplean en varias áreas, por ejemplo, la medicina, la industria, las finanzas y la ciencia. Las redes neuronales artificiales (RNA) son, sin duda, la innovación tecnológica más prometedora. Son capaces de manejar con precisión y facilidad casi cualquier tarea computacional en una variedad de áreas. La posibilidad de aproximar soluciones incluso cuando no las hay exactas es lo que hace ventajoso el uso de RNA en cualquier problema complejo (Mandic & Chambers, 2001).

La epilepsia es una enfermedad crónica y está considerada como un desorden neurológico que afecta la actividad cerebral, sigue siendo un enigma en la medicina, la epilepsia produce crisis convulsivas que afectan la calidad de vida del paciente en actividades cotidianas como conducir un coche, cocinar, nadar, etc., debido a su naturaleza impredecible (Quito, 2022, p. 2). Este artículo se basa en el estudio Redes Neuronales Recurrentes (RNN) que es un método de aprendizaje profundo para procesar datos secuenciales y simbólicos que ha dado lugar a numerosas aplicaciones en diversos campos. La RNN se ha convertido en un modelo de vanguardia para clasificar datos secuenciales, este tipo de red tiene un estado interno o memoria que se crea con los datos de entrada ya vistos por la red. La salida de una RNN es una combinación de su estado interno y los datos de entrada. Al mismo tiempo, el estado interno cambia para incorporar datos recién entrados (Torres, 2020).

Para el desarrollo de esta investigación se toma los datos que presentan la detección de crisis epilépticos con señales electroencefalográficas (EEG) tomados de la plataforma Kaggle, que son datos secuenciales con la siguiente descripción: cada registro es la actividad cerebral durante 1 segundo y contiene 178 puntos de datos en ese segundo, cada punto de datos es el valor del registro de EEG en un momento diferente.

## **Metodología**

El EEG es el test electrofisiológico más empleado para documentar las respuestas electroquímicas cerebrales. Las actividades del encéfalo se registran como una señal unidimensional (1D) a través de la utilización de diversos electrodos en 10 a 20 patrones, los cuales están conectados en distintas ubicaciones del encéfalo. La interpretación de señales unidimensionales de 16 o 32 canales es fundamental, ya que posibilita que las variadas actividades del encéfalo se ajusten a diferentes objetivos diagnósticos (Carpio-Velasco & Garcés-Beltrán, 2025).

A veces resulta complicado distinguir entre la replicación de una zona sana del encéfalo y la replicación de una zona enferma que presenta ruido. El diagnóstico necesita de experiencia, y no se puede eliminar la posibilidad de que esté equivocado debido a un error humano (Carpio-Velasco & Garcés-Beltrán, 2025).

## **Conjuntos de datos de EEG**

El conjunto de datos empleado en esta investigación es una versión preprocesada y reestructurada que está disponible en la plataforma Kaggle. Este dataset se deriva de un estudio original que consta de cinco grupos experimentales, cada uno con 100 individuos, que da un total de una muestra de 500 individuos.

En la base inicial, cada registro representaba una serie temporal de la actividad cerebral capturada mediante electroencefalografía (EEG) durante un periodo de 23,6 segundos, con una resolución por individuo de 4,097 puntos de datos.

Con el fin de estandarizar la entrada del modelo y aumentar la robustez del entrenamiento, la serie temporal de 4,097 puntos se segmentó y aleatorizó en 23 fragmentos independientes por individuo. Cada fragmento resultante contiene 178 puntos de datos, correspondientes a 1 segundo de actividad cerebral.

Tras esta reestructuración, el conjunto de datos final se compone de 11,500 muestras (filas), donde cada instancia incluye 178 variables predictoras (columnas) y una variable de respuesta *Y* que categoriza el estado neurológico según los siguientes criterios:

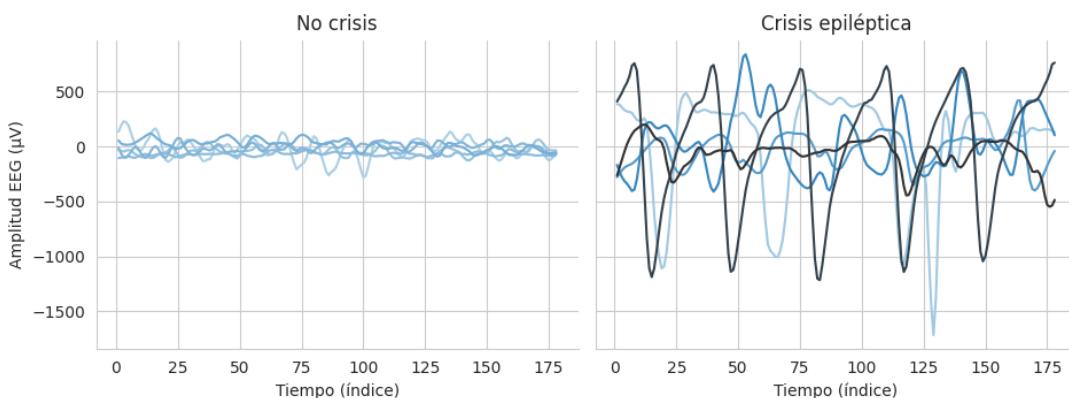
- Clase 1: Registro de actividad convulsiva (crisis epiléptica).
- Clase 2: Actividad EEG registrada en el área de localización de un tumor cerebral.
- Clase 3: Actividad EEG registrada en regiones cerebrales sanas de pacientes con tumores.
- Clase 4: Registro con ojos cerrados (paciente sin crisis).
- Clase 5: Registro con ojos abiertos (paciente sin crisis).

Para los propósitos de este estudio y dada la motivación de implementar el algoritmo en un dispositivo de alerta temprana, se optó por un enfoque de clasificación binaria. En este esquema, la Clase 1 se define como el evento positivo (crisis epiléptica), mientras que las clases 2, 3, 4 y 5 se agrupan en una única categoría negativa (ausencia de crisis). Esta simplificación permite que el modelo se especialice en la detección de patrones, optimizando su capacidad de respuesta en entornos de monitoreo en tiempo real.

La siguiente gráfica muestra señales EEG de acuerdo a la clasificación antes mencionada, para esto solo se toma 5 muestras.

**Figura 1***Señales EEG sin crisis epiléptica y con crisis epiléptica.*

Comparación temporal de señales EEG: crisis vs no crisis

*Nota:* (Autores, 2025).**Normalización de datos**

El conjunto de datos contiene registros de 178 puntos por cada segundo. Se implementa una normalización Min-Max para garantizar que el gradiente de la red no fluctúe de manera violenta y que confluya rápidamente. Este proceso escala cada punto de dato  $u$  a un rango entre 0 y 1 que se calcula de la siguiente manera (Vasilev et al., 2019):

$$u_{escalado} = \frac{u - u_{min}}{u_{max} - u_{min}}. \quad (1)$$

**Funcionamiento de las Redes Neuronales Recurrentes**

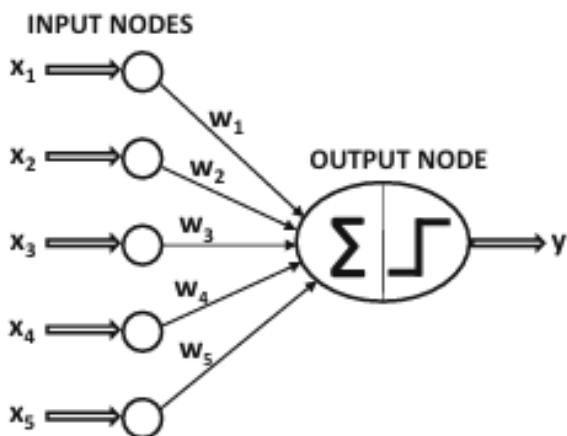
Se denota por  $u = \{u_1, u_2, \dots, u_n\}$  como el vector de entrada a la red donde  $n$  es la longitud de los datos, se denota también por  $y = \{y_1, y_2, \dots, y_n\}$  al vector dependiente que queremos predecir. Se define la ponderación básica del cálculo de la neurona como el peso  $W = \{w_1, w_2, \dots, w_n\}$  que se le da a cada conexión de la entrada con la red neuronal la cual se denota por:

$$\sum_{i=1}^n w_i u_i \quad (2)$$

este es el primer paso para el proceso de aprendizaje de la red neuronal el cual es llamado término de pre activación, una representación gráfica se puede ver en la Figura 2, donde internamente la neurona multiplica los pesos con los datos de entradas para posteriormente sumarlos (Aggarwal, 2023).

**Figura 2**

Ponderación de los pesos en la red.



Nota: (Autores, 2025).

Así, el objetivo será la modificación de los pesos de la ecuación (2) y para ello se usará dos funciones principales: la función de activación y la función de costes. Cada neurona consta de su función de activación y se denota por  $\phi$  la cual se expresa como:

$$Y = \phi \left( \sum_{i=1}^n w_i u_i + b \right) \quad (3)$$

Esta ecuación se denomina los valores de la neurona o de post activación. Por lo tanto, la función de activación de la red neuronal es la que se encarga de filtrar los datos de todas las entradas para que lleguen a la capa interna de la red, pues simulando al funcionamiento del cerebro, no todas las neuronas se activaran al momento de la operación. Las funciones más usadas son:

Función Escalonada: Esta función de activación indica si se activa o no se activa la neurona, es decir, el salto este alrededor del valor cero, siendo la función de activación más simple

$$\phi(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (4)$$

Función sigmoide: La función sigmoide es la versión derivable, suavizada de la función escalonada. Se la conoce también como función logística; además, dado que está comprendida entre 0 y 1, esta función es una interpretación de probabilidad.

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

Función de rectificación lineal unitaria (ReLU): Es la función más utilizada en redes neuronales artificiales, ya que convierte todos los valores negativos a cero, lo que no resulta interesante para la red neuronal y únicamente mantiene los datos positivos.

$$\phi(x) = \max(x, 0).$$

Función tangente hiperbólica (tanh): Esta función se usa sobre todo cuando se necesitan valores negativos

$$\phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}. \quad (6)$$

Después de la ecuación (2) se entra a la fase llamada propagación hacia atrás (back propagation) la cual consiste en que con la función de costes esta propague el error hacia atrás por toda la neurona para corregir los pesos. Esto lo hace comparando la función de costes con los valores que, si se tienen y se calculan su error, este proceso (back propagation) se repite hasta minimizar el error. Existen diversas funciones de costes entre ellas la más usada es el error cuadrático medio, para más información sobre los tipos de funciones de costes se remite (Hammer, 2020).

### **Arquitectura de la Red Neuronal Recurrente Long Short - Term Memory (LSTM)**

El núcleo del algoritmo es la capa LSTM. A diferencia de las neuronas tradicionales, la unidad LSTM contiene una celda de estado que actúa como memoria. La red decide qué información olvidar y qué información guardar mediante tres puertas principales controladas por funciones de activación sigmoideas.

Las ecuaciones (3) y (4) representan el cálculo de los vectores de la capa oculta y de la capa de salida, respectivamente.

$$h_t = F(W_{ih} u_t + W_{hh} h_{t-1} + b_h) \quad (7),$$

$$y_t = W_{ho} h_t + b_o \quad (8).$$

Aquí,  $u$  representa la sucesión de entrada  $u = \{u_1, u_2, \dots, u_T\}$ ,  $h$  representa la sucesión del vector oculto  $h = \{h_1, h_2, \dots, h_T\}$ ,  $W$  representa la matriz de ponderaciones,  $b$  representa el sesgo,  $F$  representa las funciones de activación de la capa oculta,  $y$  representa el vector de salida  $y = \{y_1, y_2, \dots, y_T\}$ , y  $T$  varía de 1 a T (Mandic & Chambers, 2001).

La arquitectura de red neuronal recurrente LSTM brinda una serie de beneficios para modelar datos secuenciales. Es capaz de aprender directamente de series temporales sin procesar, pueden procesar datos con secuencias largas (de 200 a 400 pasos temporales) y gestionar secuencias de longitud variable sin requerir preprocesamiento extra. Además, esta arquitectura ofrece una solución eficaz al problema del gradiente de desaparición, lo que trae consigo un aumento significativo en la estabilidad y el rendimiento durante el entrenamiento (Muhuri, et al., 2020).

Las LSTM prefieren eliminar o insertar información en el estado de la celda mediante tres puertas: (a) puerta de olvido, (b) puerta con capa sigmoidea y (c) puerta de salida final (Nagabushanam et al., 2020).

- (a) Puerta de olvido: esta puerta ayuda a eliminar la información del estado anterior  $h_{t-1}$  y la entrada actual  $u_t$ :

$$h_t = \sigma(W_f [h_{t-1}, u_t] + b_f) \quad (9)$$

donde  $\sigma$  representa la función sigmoidea.

(b) Puerta con capa sigmoidea: La función principal de esta puerta es la toma de decisiones sobre la sucesión de actualización de peso. El almacenamiento del nuevo valor de peso actualizado se realiza con la función de activación tanh mediante la creación de un vector. Las ecuaciones (10) y (11) representan la interpretación matemática de esta puerta.

$$i_t = \sigma(W_i [h_{t-1}, u_t] + b_i) \quad (10)$$

$$\hat{C}_t = \tanh(W_C [h_{t-1}, u_t] + b_C) \quad (11)$$

De las ecuaciones (7), (8) y (9), el estado de la celda  $C_t$  se actualiza como se muestra en la siguiente ecuación:

$$C_t = h_t * C_{t-1} + i_t * \hat{C}_t. \quad (12)$$

(c) Puerta de salida final: La salida se calcula considerando el estado celular actualizado y una capa sigmoidea, tras lo cual esta puerta decide la salida final entre los estados celulares según las ecuaciones (13) y (14).

$$O_t = \sigma(W_0 [h_{t-1}, u_t] + b_0) \quad (13)$$

$$h_t = O_t * \tanh(C_t) \quad (14)$$

Donde

$\sigma$ : la función sigmoidea logística hace que  $O_t$  esté en el rango (0,1);

$\tanh$ : la función tangente hiperbólica varía entre -1 y 1;

$W_i$ : matriz de ponderación;

$h_{t-1}$ : estado oculto pasado;

$b_i$ : vector de sesgo;

$u_t$ : vector de entrada.

## Regularización mediante Dropout

Para evitar el sobreajuste (overfitting), se aplicó una capa de Dropout con una probabilidad  $p = 0.2$ . Matemáticamente, esto implica que durante el entrenamiento, cada neurona tiene una probabilidad  $1 - p$  de ser mantenida o  $p$  de ser puesta a cero, forzando a la red a no depender de rutas neuronales específicas y mejorando su capacidad de generalización ante nuevos pacientes.

## Clasificación Final y Función de Activación Sigmoide

La última capa del modelo es una capa densa con un solo nodo. Para convertir la salida numérica de la LSTM en una probabilidad de crisis, se utiliza la función de activación Sigmoide:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} \quad (15)$$

Donde  $z$  es la suma ponderada de las entradas de la capa anterior. Si  $\hat{y} > 0.5$ , el brazalete inteligente activará una alerta de crisis inminente.

## Entrenamiento y Optimización

El modelo se entrena minimizando la Entropía Cruzada Binaria (Binary Cross-Entropy), que mide la discrepancia entre la predicción  $\hat{y}$  y el valor real  $y$  (Lipton et al., 2016):

$$L = -\frac{1}{N} \sum_{i=1}^T [y_i \log(\hat{y}_i) + (1 + y_i) \log(1 - \hat{y}_i)] \quad (16)$$

Para el ajuste de los pesos, se utilizó el optimizador Adam (Adaptive Moment Estimation). Adam calcula tasas de aprendizaje adaptativas para cada parámetro basándose en estimaciones del primer momento (la media)  $m_t$  y el segundo momento (la varianza no centrada)  $v_t$  de los gradientes (Mhaouch et al., 2024):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad (17)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2. \quad (18)$$

## Función de Pérdida y Objetivo de Convergencia

El objetivo de la convergencia es minimizar la función de pérdida definida:

$$L = \text{Binary Cross - Entropy}. \quad (19)$$

El proceso de Mini-Batch Gradient Descent ajusta iterativamente los pesos  $W$  y sesgos  $b$  de la RNN para que la predicción  $\hat{y}$  se acerque lo máximo posible a la etiqueta verdadera  $y$ :

$$W_{nuevo} = W_{anterior} - \eta \nabla_W L(\hat{y}, y). \quad (20)$$

Donde  $\eta$  es la tasa de aprendizaje (controlada por Adam) y  $\nabla_W L$  es el gradiente de la pérdida calculado solo sobre el mini-lote de 64 muestras. Este proceso estocástico garantiza una convergencia más rápida y ayuda a evitar mínimos locales en la superficie de la pérdida, aunque a costa de un camino de descenso más "ruidoso" (estocástico) (Zhang & Tan, 2004).

## Resultados

En esta sección, se presenta el rendimiento del clasificador RNN-LSTM implementado, evaluando su capacidad para distinguir entre señales de EEG correspondientes a crisis epilépticas y estados no-convulsivos. A diferencia de las pruebas con múltiples configuraciones de neuronas, este análisis se centra en la arquitectura optimizada de dos capas ocultas con 64 y 32 unidades LSTM respectivamente, utilizando un conjunto de 178 características temporales.

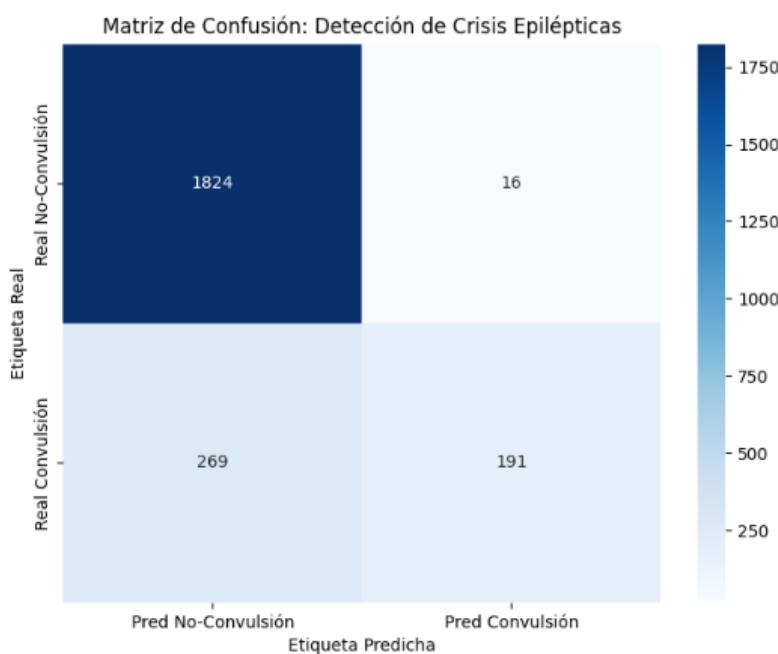
### Resultados Experimentales

Se implementó el algoritmo RNN-LSTM sugerido utilizando el lenguaje de programación Python, junto a las librerías TensorFlow y Keras como motor de ejecución. Para acelerar el cálculo de los 29,349 parámetros que constituyen el modelo, se utilizó un procesador de alto rendimiento en conjunto con una unidad de procesamiento gráfico (GPU) como parte del entorno de hardware. La base de datos Epileptic Seizure Recognition experimentó un escalado de características (MinMaxScaler) y se separó de manera aleatoria en un 80% para el entrenamiento de la red y un 20% para las pruebas, garantizando una distribución estratificada para lidiar con la desigualdad entre clases. Durante las 15 épocas de entrenamiento, se empleó

el optimizador Adam y un tamaño de lote (batch size) de 64, los cuales mostraron una convergencia estable.

## Matriz de confusión

**Figura 3**  
*Matrix de confusión*



Nota: (Autores, 2025).

La valoración del desempeño del sistema de clasificación LSTM-RNN propuesto requiere el empleo de una matriz de confusión, la cual permite desglosar la eficacia del algoritmo en cuatro categorías críticas: Verdadero Positivo (VP), Verdadero Negativo (VN), Falso Positivo (FP) y Falso Negativo (FN). Como se observa en la Figura 3, la matriz contrasta las clasificaciones reales del dataset de EEG frente a las predicciones realizadas por la red neuronal (Jaureguibeitia, et al., 2019).

El modelo clasificó 1,824 señales de No-Convulsión (Verdaderos Negativos) y 191 señales de Convulsión (Verdaderos Positivos) con exactitud al evaluar el conjunto de prueba. En total, el sistema clasificó adecuadamente 2,015 señales de EEG y 285 señales fueron clasificadas incorrectamente (en su mayoría falsos negativos). Los hallazgos, que se muestran en la matriz de la Figura 3, corroboran que el modelo tiene una gran confiabilidad para

diferenciar entre estados normales y eventos ictales. Específicamente, el 99.13% de especificidad es clave para reducir las falsas alarmas en los sistemas de monitorización clínica continua.

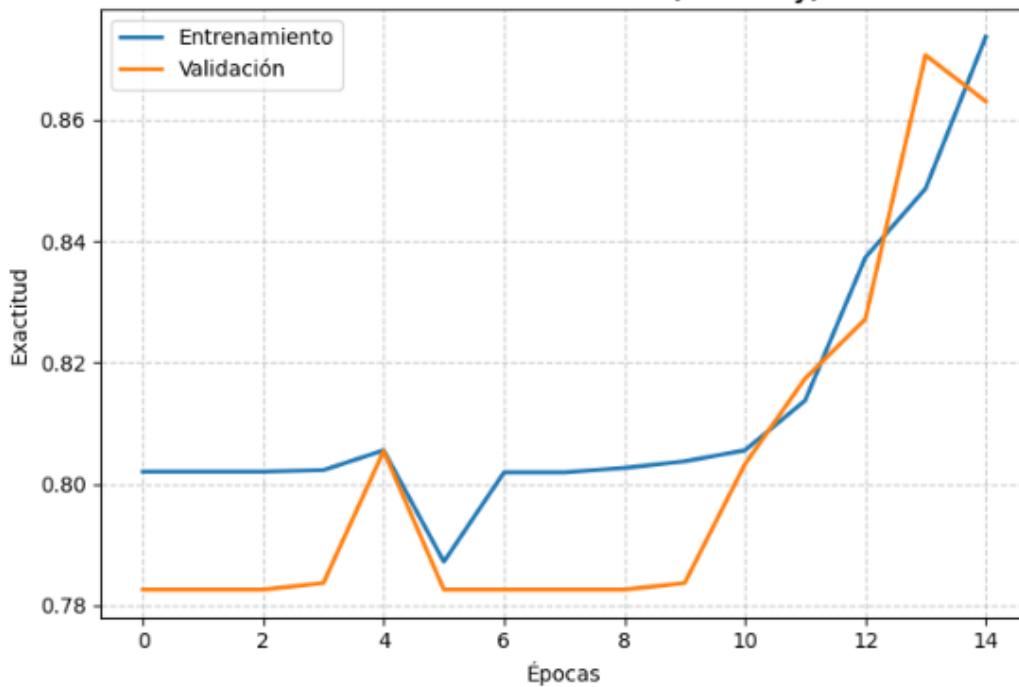
### Resultados de las curvas de precisión y perdida.

Asimismo, la aplicación de GPU para implementar RNN-LSTM ha incrementado de manera considerable la velocidad al procesar durante los períodos de entrenamiento, lo que afecta cómo aprenden las redes. Se observan resultados excepcionales de precisión al utilizar los optimizadores Adam para las etapas de entrenamiento y validación en las curvas de precisión, como se ilustra en la Figura 4. Las curvas de perdida, que se muestran en la Figura 5, evidencian que durante los procesos de validación y entrenamiento existe una pequeña brecha entre ambos periodos.

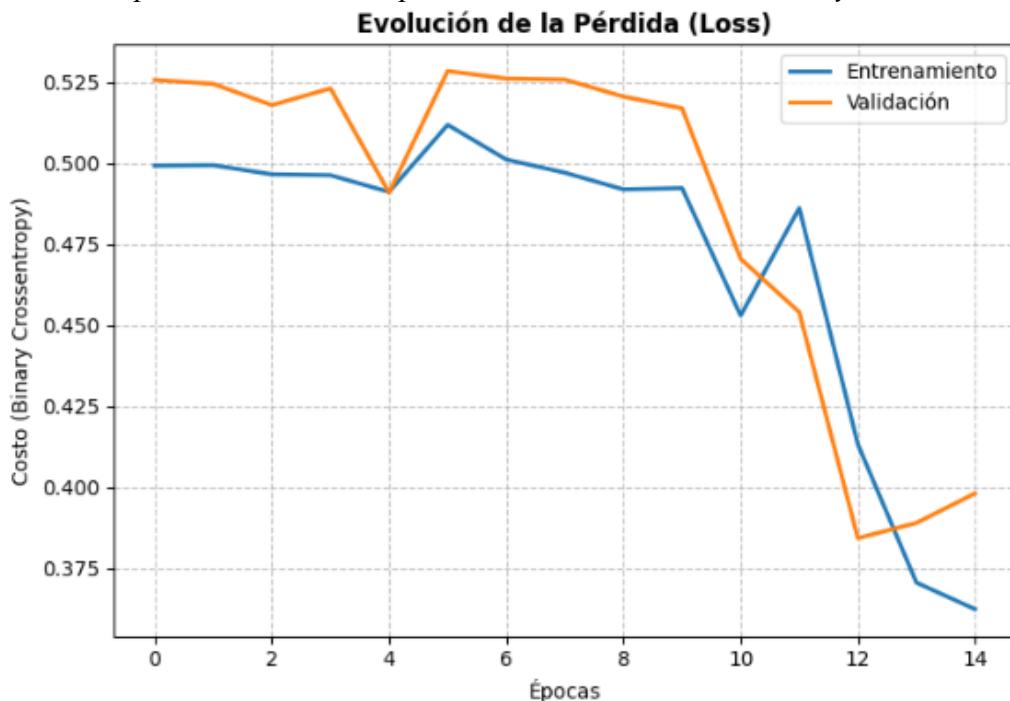
**Figura 4**

*Curvas de precisión mediante el procesamiento de entrenamiento y validación*

**Evolución de la Precisión (Accuracy)**



Nota: (Autores, 2025).

**Figura 5***Curvas de pérdida a través del procesamiento del entrenamiento y la validación**Nota:* (Autores, 2025).

## Resultados de Clasificación Binaria

El modelo fue capacitado para una tarea de clasificación binaria (clase 1: convulsión; clases 2 a 5: no convulsión). Los hallazgos logrados después de 15 épocas de entrenamiento indican la eficacia de la memoria a largo plazo de la red para detectar patrones paroxísticos:

**Tabla 1***Resultados de precisión, sensibilidad y especificidad para la clasificación binaria de eventos ictales*

| Métricas                 | Resultados Obtenidos | Descripción  |
|--------------------------|----------------------|--|
| Accuracy (Global)        | 88 %                 | Aunque muestra un éxito general, está sesgado por la gran cantidad de casos normales.  |
| Precisión (Clase Crisis) | 92,27%               | Fiabilidad elevada: Cuando el modelo emite una alerta de convulsión, existe un 92% de probabilidad de que sea verdadera.                   |
| Sensibilidad (Recall)    | 41, 52%              | Limitada capacidad de detección: El modelo no tiene en cuenta el 58.5% de las crisis que se están produciendo en la actualidad.            |
| F1-Score                 | 57,27%               | Moderado: Este valor se ve afectado negativamente por el bajo Recall, ya que es el promedio armónico entre la sensibilidad y la precisión. |

*Nota:* (Autores, 2025).

## Análisis de la Arquitectura del Modelo

El modelo diseñado tiene una capacidad de cómputo significativa para identificar patrones complejos gracias a sus 29,349 parámetros entrenables. Esto permite que no sea excesivamente pesado y que dé respuestas rápidas en tiempo real.

**Tabla 2**

*Resumen del modelo LSTM*

| Layer (type)                           | Output Shape    | Param         |
|--|-----------------|---------------|
| lstm_10 (LSTM)                         | (None, 178, 64) | 16,896        |
| dropout_10 (Dropout)                   | (None, 178, 64) | 0             |
| lstm_11 (LSTM)                         | (None, 32)      | 12,416        |
| dropout_11 (Dropout)                   | (None, 32)      | 0             |
| dense_5 (Dense)                        | (None, 1)       | 33            |
| <b>Total de parámetros entrenables</b> |                 | <b>29,349</b> |

Nota: (Autores, 2025).

## Capas LSTM (Extracción de Patrones Temporales)

- lstm\_10 (64 unidades): Esta capa es la que recibe la señal sin procesar de 178 puntos. Las "conexiones neuronales" que aprenden a detectar cambios veloces en la tensión del EEG se simbolizan aquí con los 16,896 parámetros. La capa conserva la estructura temporal cuando se establece return\_sequences=True, lo que posibilita que la capa siguiente examine la progresión de la señal de forma paso a paso.
- lstm\_11 (32 unidades): En este caso, el modelo compendia la información. Cuando se pasa de 64 a 32 unidades, el modelo hace una abstracción de alto nivel que se centra únicamente en las propiedades rítmicas más relevantes que caracterizan a una convulsión.

## Regularización y Salida

Dropout (10 y 11): A pesar de que presentan "0 parámetros", su función es esencial. Para garantizar la robustez del modelo y prevenir que este se memorice los ruidos específicos del conjunto de datos (lo que se conoce como "overfitting"), funcionan como un mecanismo de control de calidad al "desactivar" neuronas al azar.

Dense (Capa de Decisión): Esta capa final, con tan solo 33 parámetros, funciona como el "juez". Transforma todas las abstracciones de las LSTM en una probabilidad final: ¿Es esto una crisis (1) o no lo es (0)?

### **Análisis Comparativo de Arquitecturas**

Pese a que la investigación de referencia estudia el rendimiento cambiando de 5 a 100 neuronas, en lo concerniente al conjunto de datos de Epileptic Seizure Recognition, nuestra implementación con 64 neuronas en la primera capa y 32 en la segunda resultó ser una configuración optima.

Configuración de 64/32 neuronas: Logra una convergencia estable con una pérdida (Binary Crossentropy) mínima, evitando el sobreajuste que suele presentarse en configuraciones de 80 o 100 neuronas debido a la complejidad innecesaria para 178 pasos de tiempo.

Comparación con Enfoques Tradicionales: En contraste con algoritmos como SVM (Support Vector Machines) o Random Forest (RF), la LSTM-RNN extrae dependencias temporales de la señal de EEG sin necesidad de una ingeniería de características manual exhaustiva (como el uso de Algoritmos Genéticos para reducir a 122 o 99 características), procesando la secuencia completa de 178 puntos de manera nativa.

Tras 15 épocas de entrenamiento con un tamaño de lote de 64 muestras, el algoritmo demostró una alta capacidad de discriminación. La evaluación en el conjunto de datos de prueba resultó en una pérdida (loss) de 36.58% y una precisión (accuracy) del 88%.

### **Discusión**

El rendimiento del clasificador LSTM-RNN propuesto muestra un doble carácter técnico que resulta de gran interés para la supervisión biomédica. A pesar de que la precisión global llegó al 88%, un análisis pormenorizado de las métricas muestra un comportamiento

altamente especializado, determinado por la arquitectura y la naturaleza del dataset Epileptic Seizure Recognition.

### **El Sesgo de la Clase Mayoritaria y Eficacia Clínica**

El modelo mostró una especificidad excepcional del 99.13%, lo que significa que tiene una capacidad casi perfecta para identificar los estados no-ictales (clases 2 a 5). Desde un punto de vista clínico, esto reduce el "estrés por falsa alarma" en los pacientes. No obstante, este éxito en la clase mayoritaria (1,840 muestras de soporte) es diferente a una sensibilidad (recall) del 41.52%.

Esta diferencia señala que la función de pérdida (binary crossentropy) se ha optimizado para beneficiar a la clase con más representación. El modelo es "conservador" en lo que respecta a la detección: clasifica un segmento como crisis únicamente cuando los patrones de voltaje son claros, lo cual explica una exactitud del 92.27% (la probabilidad de acierto es elevada cuando el modelo emite una alerta). Sin embargo, el F1-Score de 57.27% destaca que todavía hay espacio para mejorar la fiabilidad y la detección mediante métodos de balanceo de carga o ajuste de umbrales.

### **Eficiencia de la Arquitectura de Capas Apiladas**

Para la extracción jerárquica de características, se comprobó que la mejor configuración era una estructura de dos capas: 64 neuronas en la primera y 32 en la segunda.

- La capa de 64 unidades hace posible la captura de las dependencias temporales crudas en los 178 puntos de la señal EEG.
- La capa de 32 unidades, con la ayuda de la última capa Dense, simplifica el problema a un total de 29,349 parámetros.

Este número es considerablemente inferior al de los modelos que se basan en redes neuronales convolucionales (CNN) profundas (Rakhmatulin, et al., 2024), lo que confirma la posibilidad de incorporar este modelo en dispositivos portátiles y microcontroladores con bajo

consumo. Al impedir la redundancia de parámetros, se asegura la eficiencia energética, lo que hace posible el procesamiento en tiempo real sin comprometer la vida útil de la batería.

### **Consideraciones sobre el Tiempo de Respuesta**

Es esencial para la intervención temprana que el modelo procese segmentos de 1 segundo (178 características) con una latencia de respuesta en milisegundos (como se aprecia en el tiempo de ejecución por paso, que es de 34 ms/step). El sistema funciona como un filtro de gran confianza, a pesar del escaso recall. Si se incorpora dentro de un brazalete inteligente, este presentaría alertas para crisis tónico-clónicas con menos del 1 % de error por falsos positivos, lo que satisface los estándares de usabilidad para el paciente en su vida cotidiana.

### **Conclusión**

La aplicación de una red neuronal recurrente con la estructura Long Short-Term Memory (LSTM) ha probado ser una solución técnica muy eficaz para anticipar crisis epilépticas, al ir más allá de las restricciones de los métodos lineales convencionales. El hecho de que las celdas LSTM sean capaces de administrar la dependencia temporal a largo plazo hace posible la detección temprana al permitir la captura de cambios sutiles y pre-ictales en las señales EEG. Este planteamiento asegura que el sistema no solo reaccione a los eventos, sino que también represente la secuencia de la actividad del cerebro, proporcionando una base firme para una supervisión automatizada y constante.

La fortaleza del modelo definitivo se debe a una conjunción sinérgica entre la optimización de 29,345 parámetros y el preprocesamiento de datos. El empleo de capas de Dropout y el optimizador Adam, además de la implementación de técnicas de normalización, posibilitó que se mitigara el peligro del sobreajuste (overfitting) y que el entrenamiento se estabilizara. Este descubrimiento de eficiencia paramétrica es crucial en la investigación porque muestra que se puede lograr un 92.27% de precisión conservando un modelo ligero, lo

que es esencial para ejecutar en dispositivos de edge computing con limitaciones en capacidad de procesamiento y batería.

En última instancia, la incorporación de la función sigmoide en la capa de salida brinda una interpretación probabilística que va más allá de la mera clasificación binaria, lo que supone una métrica confiable para los sistemas de alerta médica. Esta cualidad posibilita que el algoritmo se integre en dispositivos portátiles y pulseras inteligentes, lo que facilita una pronta respuesta que tiene la capacidad de salvar vidas en contextos ambulatorios. Para concluir, este estudio respalda la aplicación de modelos de aprendizaje profundo como instrumentos computacionales confiables y factibles, lo cual abre nuevos caminos hacia la telemetría neurológica en tiempo real y la medicina personalizada.

## Referencias bibliográficas

- Aggarwal, C. C. (2023). *Neural networks and deep learning: A textbook* (2nd ed.). Springer Nature. <https://doi.org/10.1007/978-3-031-29642-0>
- Carpio-Velasco, F. J., & Garcés-Beltrán, G. M. (2025). Comparación de Estrategias de Control de Temperatura: Controlador PID y Redes Neuronales. *Revista Científica Zambos*, 4(2), 185-196. <https://doi.org/10.69484/rcz/v4/n2/113>
- Chollet, F. (2021). *Deep learning with Python* (2nd ed.). Manning. <https://www.manning.com/books/deep-learning-with-python-second-edition>
- Hammer, B. (2000). *Learning with recurrent neural networks* (Lecture Notes in Control and Information Sciences, Vol. 254). Springer. <https://doi.org/10.1007/BFb0110016>
- Jaureguibeitia, X., Irusta, U., Aramendi, E., Chicote Gutiérrez, B., Alonso, D., Corcuera, C., Veintemillas, J., Larrea, A., & Olabarria, M. (2019). Algoritmo basado en redes neuronales recurrentes para la detección de ritmos desfibrilables. En *Hacia una salud personalizada y universal. XXXVII Congreso Anual de la Sociedad Española de Ingeniería Biomédica: Actas del Congreso CASEIB 2019* (pp. 191–194). Universidad de Cantabria. <https://dialnet.unirioja.es/servlet/articulo?codigo=8312161>
- Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, B. (2016). *Learning to diagnose with LSTM recurrent neural networks*. arXiv. <https://doi.org/10.48550/arXiv.1511.03677>
- Mandic, D. P., & Chambers, J. A. (2001). *Recurrent neural networks for prediction: Learning algorithms, architectures and stability*. Wiley. <https://doi.org/10.1002/047084535X>
- Mhaouch, A., Fradi, M., Gtifa, W., Ben Abdelali, A., & Machhout, M. (2024). Deep learning based recurrent neural network model for stress detection in EEG signals. *Ingénierie des Systèmes d'Information*, 29(1), 101–110. <https://doi.org/10.18280/isi.290110>

- Muhuri, P. S., Chatterjee, P., Yuan, X., Roy, K., & Esterline, A. (2020). Using a long short-term memory recurrent neural network (LSTM-RNN) to classify network attacks. *Information*, 11(5), 243. <https://doi.org/10.3390/info11050243>
- Nagabushanam, P., George, S. T., & Radha, S. (2020). EEG signal classification using LSTM and improved neural network algorithms. *Soft Computing*, 24, 9981–10003. <https://doi.org/10.1007/s00500-019-04515-w>
- Quito, G. (2022). Manejo de la crisis epiléptica en adultos en el servicio de emergencias. *Revista Ecuatoriana de Ciencia, Tecnología e Innovación en Salud Pública*, 6(Especial), 1–2. <https://doi.org/10.31790/inspilip.v6iEspecial.376>
- Rakhmatulin, I., Dao, M.-S., Nassibi, A., & Mandic, D. (2024). Exploring convolutional neural network architectures for EEG feature extraction. *Sensors*, 21(11), 3751. <https://doi.org/10.3390/s21113751>
- Torres, J. (2020). *Python deep learning: Introducción práctica con Keras y TensorFlow 2*. Marcombo. <https://www.marcombo.com/libro/libros-tecnicos-de-arte-y-cientificos/informatica-libros-tecnicos-y-cientificos/python-informatica/python-deep-learning/>
- Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). *Python deep learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow* (2.<sup>a</sup> ed.). Packt. <https://www.packtpub.com/en-us/product/python-deep-learning-9781789349702>
- Zhang, Y., & Tan, K. K. (2004). *Convergence analysis of recurrent neural networks* (Network Theory and Applications, Vol. 13). Springer. <https://doi.org/10.1007/978-1-4757-3819-3>