






Pronóstico de viento en la región del Istmo mediante reconstrucción parcial de *wavelets* y aprendizaje automático

Wind speed forecasting in the Isthmus region using partial wavelet reconstruction and machine learning

Brandon N. Nava-Martínez¹ , José L. Martínez-Rodríguez^{1,*} , Ana B. Ríos-Alvarado¹ , Tania Y. Guerrero-Melendez¹ 
, Benigno Estrada-Drouaillet¹ , Juan Carlos Elizondo-Leal¹ 

¹Facultad de Ingeniería y Ciencias. Supermanzana Campus, Centro Universitario, 87149 Cd. Victoria, Tamps.
a2163010063@alumnos.uat.edu.mx (B.N.N.-M); arios@docentes.uat.edu.mx (A.B.R.-A); tyguerre@docentes.uat.edu.mx
(T.Y.G.-M); benestrada@docentes.uat.edu.mx (B.E.-D); jcaelizondo@docentes.uat.edu.mx (J.C.E.-L); *Autor para
correspondencia: lazaro.martinez@uat.edu.mx

RESUMEN: La predicción precisa de la velocidad del viento es importante para optimizar la generación de energía eólica, ya que mejora la planificación del suministro eléctrico y el mantenimiento de los aerogeneradores. Este trabajo propone un modelo híbrido que integra la Transformada Wavelet Discreta (DWT por sus siglas en inglés) con redes neuronales LSTM para el pronóstico a corto plazo. Se utilizó una serie temporal de datos de velocidad de viento de Nizanda, Oaxaca, aplicando la wavelet Daubechies 4 (db4) para descomponer la señal en componentes de aproximación y detalles. Se evaluaron múltiples configuraciones de reconstrucción parcial para filtrar el ruido y resaltar patrones relevantes. Los resultados experimentales muestran que la reconstrucción selectiva utilizando la aproximación y los primeros dos primeros niveles de detalle, no solo supera al modelo base, el cual consiste en una LSTM alimentada con datos de velocidad de viento sin transformada wavelet, sino también a los modelos tradicionales de referencia (persistencia, ARIMA, SARIMA). Esta configuración alcanzó un R^2 de 0.934 y un RMSE de 0.979 en el conjunto de prueba.

PALABRAS CLAVES: Pronóstico de viento, aprendizaje automático, transformaciones wavelet, energía eólica, series de tiempo.

ABSTRACT: Accurate wind speed prediction is important for optimizing wind energy generation, as it improves power supply planning and the maintenance of wind turbines. This paper proposes a hybrid model that integrates the Discrete Wavelet Transform (DWT) with LSTM neural networks for short-term forecasting. A time series of wind speed data from Nizanda, Oaxaca, was used, applying the Daubechies 4 (db4) wavelet to decompose the signal into approximation and detail components. Multiple partial reconstruction configurations were evaluated to filter noise and highlight relevant patterns. Experimental results demonstrate that selective reconstruction using the approximation and the first two levels of detail outperforms the base model, which consists of an LSTM with raw wind speed data without the wavelet transformation, but also traditional reference models (persistence, ARIMA,

SARIMA). This configuration achieved an R^2 of 0.934 and an RMSE of 0.979 on the test set.

KEYWORDS: Wind forecasting, machine learning, wavelet transformation, wind energy, time series

INTRODUCCIÓN

Durante las últimas dos décadas, la generación de energía renovable mediante modelos de parques eólicos ha tenido mayor impulso en el mundo (Azamar Alonso & García Beltrán, 2021). Con el crecimiento de la capacidad instalada y el número de parques eólicos, surge la necesidad de herramientas avanzadas que permitan predecir de manera precisa la generación de energía. Estas predicciones son cruciales para planificar y optimizar el aprovechamiento de los recursos eólicos, especialmente ante la variabilidad y complejidad de los patrones de viento que afectan directamente la generación eléctrica (Cano Torres & Rodríguez Cruz, 2020; Eólica, 2015; Wang *et al.*, 2021).

Las series de velocidad del viento se caracterizan por su alta variabilidad, complejidad e incertidumbre asociada a los instrumentos utilizados en las mediciones. Esta combinación de factores introduce ruido en los registros y dificulta el perfeccionamiento de los modelos de pronóstico.

Para mitigar estos efectos, diversos modelos de predicción han sido propuestos e implementados en el dominio de la energía eólica (Guo *et al.*, 2023; Wang *et al.*, 2021; Yu *et al.*, 2025). Estos modelos no solo consideran predicciones utilizando datos históricos, ya sea directamente de la energía generada, sino que también emplean datos históricos de la velocidad del viento como variable indirecta, transformándolos para calcular su equivalencia en energía. En este sentido, se puede utilizar la velocidad del viento como una señal que se traduce en patrones energéticos, lo que mejora la precisión de las predicciones (Wang *et al.*, 2021).

Por lo tanto, dado que los datos de viento suelen ser tomados como una señal, se pueden utilizar estrategias matemáticas para su análisis, por ejemplo, utilizando la transformada wavelet, las cuales permiten descomponer la señal en múltiples escalas,

capturando tanto fluctuaciones a corto plazo como tendencias a largo plazo (Soman, 2010). En el transcurso del tiempo se ha utilizado esta transformada wavelet con diferentes enfoques como en el trabajo presentado por K U and Kovoov (2021) en la cual hace las predicciones de los coeficientes para cada nivel de descomposición y reconstruye para obtener las predicciones finales. Por otro lado, Domínguez-Navarro *et al.* (2021) usan la transformada wavelet como filtro, y en Kio *et al.* (2024) usan los coeficientes obtenidos para agregarlos como características a un modelo de red neuronal. Aunque el uso de la transformada wavelet ha sido ampliamente estudiado en la predicción de series de tiempo, hasta donde abarca la revisión realizada, no se ha explorado la reconstrucción parcial como estrategia de filtrado para mejorar el desempeño de modelos LSTM, ni la aplicación de Hyperband para la optimización de hiperparámetros en la región del Istmo.

En este artículo se propone una estrategia para pronosticar la velocidad del viento a corto plazo utilizando datos del Istmo. Esta estrategia integra muestras de datos mediante la Transformada Wavelet Discreta (DWT) y, posteriormente, realiza una predicción utilizando redes neuronales LSTM. La propuesta se basa en una técnica de reconstrucción parcial selectiva de la serie de tiempo (enfoque de filtro), en la que se evalúan diferentes combinaciones de los componentes de Aproximación y Detalles de una DWT. Posteriormente, se realiza una comparación del rendimiento de la transformada wavelet con reconstrucción parcial frente a un modelo de referencia que no utiliza dicha transformación.

Fundamentos y trabajo relacionado

En esta sección se presentan los fundamentos teóricos necesarios para comprender los enfoques propuestos. Se describen los principios de la transformada wavelet y las redes LSTM, así como los principales enfoques de aplicación de transformada wavelet en predicción de series de tiempo.

Transformada Wavelet

La Transformada Wavelet es una herramienta que permite descomponer una señal en componentes de

aproximación y detalle. Mientras que la aproximación conserva la información sobre la tendencia, los niveles de detalle capturan las fluctuaciones de la señal (Vishnutheerth *et al.*, 2024). A diferencia de la transformada de Fourier, esta técnica permite realizar un análisis multi-resolución, es decir, los componentes de baja frecuencia se describen con alta resolución en frecuencia, pero con baja resolución temporal, mientras que los componentes de alta frecuencia se describen con alta resolución temporal, pero con baja resolución en frecuencia (Castro, 1995). Para esto, utiliza una descomposición jerárquica de la señal con un conjunto de coeficientes que representan tendencias (aproximaciones) y fluctuaciones (detalles) en distintos niveles de escala. En la Figura 1 se muestra el funcionamiento de la transformada wavelet discreta. Los recuadros situados a la izquierda (en verde) representan las aproximaciones, mientras que los recuadros restantes (en naranja) corresponden a los niveles de detalle.

- El parámetro n controla la localización temporal, indicando la posición donde se aplica la wavelet.

La transformada wavelet discreta de una función $f(x)$ se obtiene calculando los productos internos entre la señal y las funciones de la base como se muestra en la Ecuación 2.

$$W(m, n) = \int_{-\infty}^{+\infty} \psi_{m,n}(x) f(x) dx = \langle \psi_{m,n}, f \rangle \quad (2)$$

Los coeficientes $W(m, n)$ representan la contribución de la señal en cada escala y posición. En la práctica, estos coeficientes se dividen en:

- Coeficientes de aproximación (A_j): resumen la tendencia global de la señal.
- Coeficientes de detalle (D_j): capturan las variaciones rápidas o locales.

Enfoque de filtrado con transformada wavelet discreta

La transformada wavelet discreta tiene como característica la capacidad de realizar reconstrucciones selectivas o parciales de la señal original. Después de la descomposición, la señal se divide en un componente de aproximación (coeficientes A_j) y una serie de componentes de detalle (D_j), donde J es el máximo nivel de descomposición y $j= 1, 2, \dots, J$ identifica cada uno de los niveles de detalle.

La reconstrucción completa de la señal se logra mediante la combinación de todos los coeficientes de aproximación del nivel más alto (A_J) y todos los coeficientes de detalle de los niveles J a 1. Sin embargo, para el enfoque de filtrado se reconstruye la señal utilizando solo un subconjunto de estos coeficientes. Por lo tanto, la reconstrucción parcial se enfoca en la capacidad de los coeficientes de detalle para aislar bandas de frecuencia específicas. Los componentes de detalle (D_j) contienen la información de alta frecuencia de la señal, mientras que el componente de aproximación (A_j) contiene la información de baja frecuencia (también llamada información o tendencia).

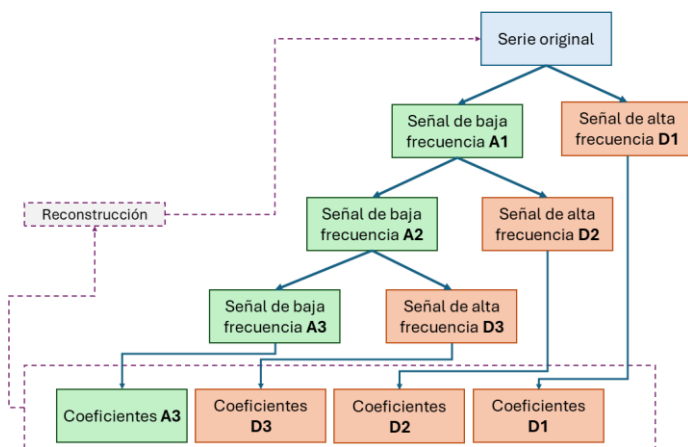


Figura 1. Ejemplo del proceso de transformada wavelet discreta.

Matemáticamente, la DWT se fundamenta en la construcción de una base ortonormal a partir de una wavelet madre $\psi(x)$, que se dilata y traslada para generar funciones de análisis como se muestra en la Ecuación 1.

$$\psi_{m,n}(x) = 2^{-\frac{m}{2}} \psi(2^{-m}x - n), \quad m, n \in \mathbb{Z} \quad (1)$$

- El parámetro m controla la escala, determinando el ancho de la wavelet.

Familias Wavelet

Existen una gran variedad de wavelets. Las propiedades fundamentales de una wavelet son su región de soporte y el número de momentos nulos (Domínguez-Navarro *et al.*, 2021). Aunque la literatura ofrece diversas familias de wavelets como Haar, Coiflet o Symlet, en este trabajo se optó por utilizar la familia Daubechies (db4) debido a sus propiedades de ortogonalidad y a diferencia de wavelets más simples como la Haar, la db4 presenta una mayor regularidad y cuatro momentos nulos (Daubechies, 1992). En la Figura 2 se pueden apreciar las funciones wavelet, las cuales fueron obtenidas gracias a la biblioteca PyWavelets (Lee *et al.*, 2019).

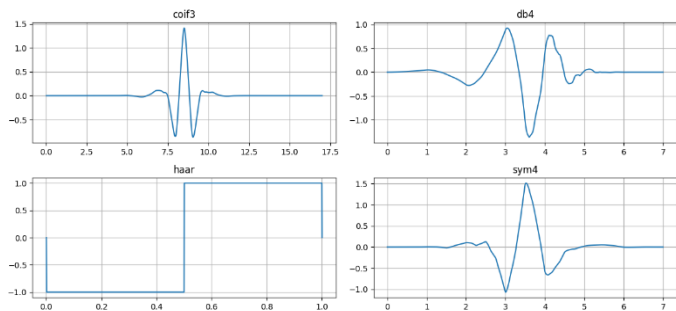


Figura 2. Ejemplo de funciones wavelet

En la literatura reciente, diversos estudios han explorado el uso de la transformada wavelet como herramienta para mejorar la predicción de la velocidad de viento y la generación de energía eólica. Por ejemplo, Nascimento *et al.* (2023) plantea una arquitectura de red neuronal profunda basada en *transformer* integrada con la transformada wavelet para la predicción de velocidad de viento y la generación de energía eólica. Por otro lado, Kio *et al.* (2024) introduce el método *Wavelet Decomposition-Neural Networks* (WDNN), que combina los principios de la descomposición wavelet con el aprendizaje profundo. Además, Amirteimoury *et al.* (2025) propone la combinación de la transformada wavelet discreta, la información mutua, el algoritmo de optimización Coot y la Red Bidireccional Long Short-Term Memory (BiLSTM) para capturar patrones complejos de la velocidad de viento.

Redes Long Short-Term Memory (LSTM)

Las redes LSTM, una forma especializada de redes neuronales recurrentes (RNN), son especialmente eficaces para manejar conjuntos de datos grandes y

complejos, gracias a su capacidad para capturar dependencias temporales de largo alcance, lo que las hace idóneas para el análisis de series temporales (Li & Law, 2024). En las RNN, la unidad recurrente se define mediante una simple activación *tanh*, mientras que, en las LSTM, esta adopta una arquitectura más elaborada y compleja. Para comprender mejor la celda LSTM, en la Figura 3 se presenta un esquema de su arquitectura, cuyos componentes se describen a continuación.

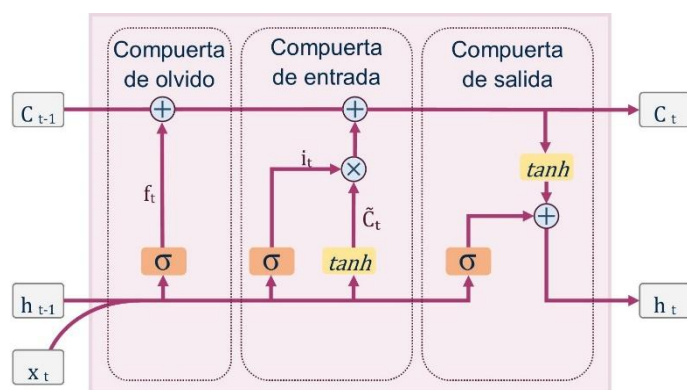


Figura 3. Celda LSTM

Estado de celda: Dentro de la red, el estado de la celda se propaga mientras la información es añadida o descartada de manera selectiva mediante mecanismos de compuertas (*gates*). Una unidad LSTM incorpora tres compuertas: la compuerta de olvido (*forget gate*), la compuerta de entrada (*input gate*) y la compuerta de salida (*output gate*).

Compuerta de olvido: Una capa sigmoide se incorpora para decidir qué información debe descartarse del estado de la celda. Esta decisión se basa en la combinación de la entrada actual x_t y la salida previa h_{t-1} de la unidad LSTM, como se muestra en la Ecuación 3.

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f) \quad (3)$$

Donde W^f y U^f son los pesos asociados con x_t y h_{t-1} , respectivamente, mientras que b^f representa el vector de sesgo de la compuerta de olvido.

Compuerta de entrada: La estructura consta de dos capas: una capa sigmoide que regula las decisiones de actualización, seguida de una capa *tanh* que genera un vector de valores candidatos, denotado como \tilde{C}_t , el

cual puede integrarse en el estado de la celda. Esto se muestra en la Ecuación 4 y Ecuación 5, respectivamente.

$$i_t = \sigma(W^i \cdot x_t + U^i h_{t-1} + b^i) \quad (4)$$

$$\tilde{C}_t = \tanh(W^c \cdot x_t + U^c h_{t-1} + b^c) \quad (5)$$

Donde W^i y U^i son los pesos asociados con x_t y h_{t-1} , respectivamente, mientras que b^i representa el vector de sesgo de la compuerta de entrada.

Para calcular el estado de celda actualizado C_t , el modelo primero atenúa el estado previo C_{t-1} mediante la compuerta de olvido f_t , y posteriormente integra nueva información al escalar el estado candidato \tilde{C}_t con la activación de la compuerta de entrada, esto se expresa en la Ecuación 6.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (6)$$

Compuerta de salida: La compuerta de salida regula el cálculo del siguiente estado oculto h_t , el cual codifica información proveniente de entradas anteriores. El proceso comienza aplicando una función sigmoide a la entrada actual y al estado oculto previo, seguido de una transformación \tanh del estado de celda actualizado C_t , como se muestra en la Ecuación 7 y Ecuación 8 respectivamente.

$$O_t = \sigma(W^o x_t + U^o h_{t-1} + b^o) \quad (7)$$

$$h_t = O_t * \tanh(C_t) \quad (8)$$

Donde W^o y U^o son los pesos asociados con x_t y h_{t-1} , respectivamente, mientras que b^o representa el vector de sesgo de la compuerta de salida.

Diversos autores han abordado el potencial de las redes LSTM como técnica para mejorar la precisión en la predicción de velocidad del viento y en la estimación de la energía eólica. Por ejemplo, K U and Koor (2021) plantea la combinación de la transformada wavelet, las redes LSTM y la Regresión por Vectores de Soporte (SVR). Por otro lado,

Buratto *et al.* (2024) proponen el uso de la transformada wavelet y las redes neuronales convolucionales (CNN) para extraer características, y las redes LSTM para realizar las predicciones. Además, Tamilselvi *et al.* (2024) presentan un algoritmo novedoso que combina la transformada wavelet con redes LSTM.

Aunque se han realizado avances que utilizan ambas herramientas (transformada wavelet y redes LSTM), es conveniente proponer una solución que aproveche las capacidades de reconstrucción de las wavelets para preparar un modelo que sea capaz de mitigar señales abruptas y, con ello, mejorar los resultados de predicción. Además, este tipo de estrategias aún no han sido aplicadas a los datos de la región Istmo, lo que alienta a conocer su comportamiento en dicho contexto.

METODOLOGIA

La metodología desarrollada, cuya implementación se realizó en el lenguaje de programación Python, se estructuró en cuatro etapas, como se muestra en la Figura 4. El flujo de trabajo inicia con la obtención de datos climáticos de alta resolución, seguido de un preprocesamiento. Posteriormente, se aplica la descomposición wavelet para filtrar y reconstruir señales en Aproximación (Aprox) y componentes de Detalle (D1, D2, D3). Con esta información, se entrena una red LSTM optimizada mediante técnicas de ajuste de hiperparámetros. En la etapa final, se evalúa y compara el desempeño de los diferentes modelos entrenados con el fin de identificar la configuración de mayor rendimiento. En la Figura 6 se muestra el pseudocódigo general de la propuesta de filtrado wavelet-LSTM.

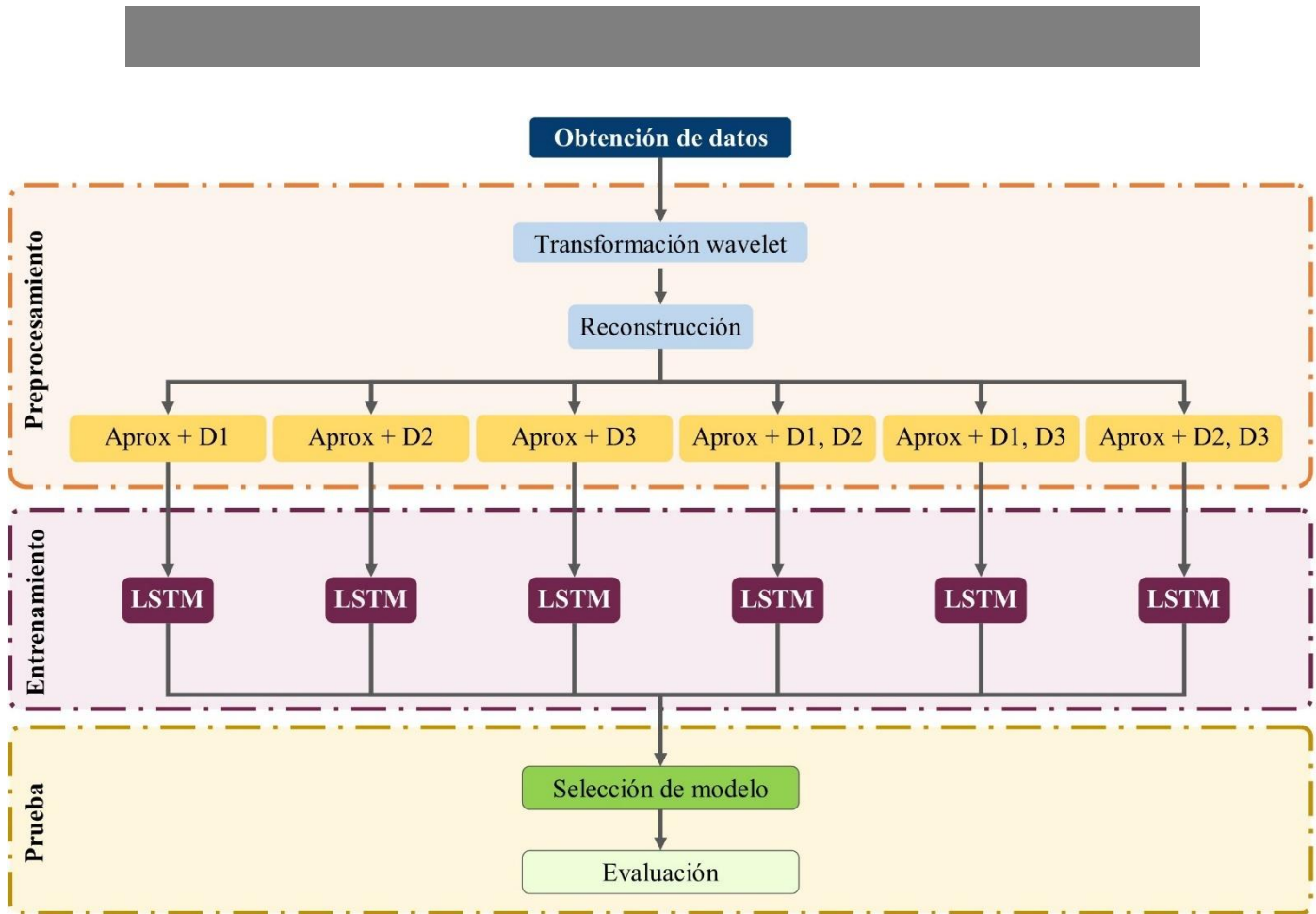


Figura 4.- Metodología general

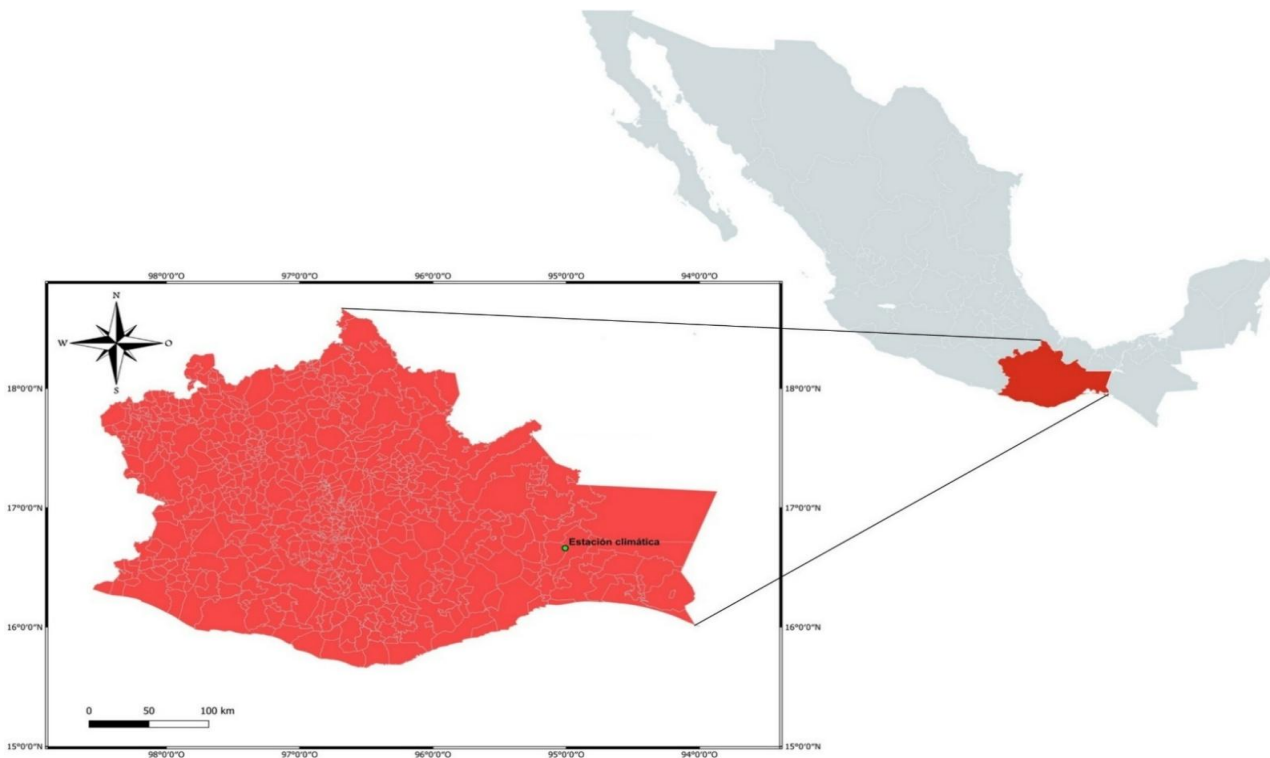


Figura 4. Ubicación de la estación meteorológica

Requiere: Serie de tiempo $S = \{s_1, s_2, \dots, s_n\} \subset \mathbb{R}$.

Parámetros: L = nivel de descomposición, p = proporción de prueba, $window$ = tamaño de ventana, h = horizonte de predicción.

- 1: Dividir serie en entrenamiento y prueba:**
 $S_{train} = \{s_1, \dots, s_m\}$ // fragmento de entrenamiento
 $S_{test} = \{s_{m+1}, \dots, s_n\}$
- 2: Aplicar DWT a S_{train} hasta nivel L :**
 $W(S_{train}) = \{A_L, D_1, D_2, \dots, D_L\}$
- 3: Para cada combinación de coeficientes:**
 $C^k = A_L + \sum\{i \in D_L\} D_i$
 $\hat{S}^k = W^{-1}(C^{(k)})$ // reconstrucción filtrada
- 4: Escalar datos:**
Ajustar MinMaxScaler con \hat{S}_{train}^k
Transformar \hat{S}_{train}^k y \hat{S}_{test}^k
- 5: Crear secuencias (X, y) con ventana y horizonte:**
 $X_{tr}^k, Y_{tr}^k = create_sequences(\hat{S}_{train}^k, window, h)$
 $X_{te}^k, Y_{te}^k = create_sequences(\hat{S}_{test}^k, window, h)$
- 6: Optimización de hiperparámetros:**
Ejecutar Hyperband sobre X_{tr}^k, Y_{tr}^k
Seleccionar hiperparámetros óptimos θ^k
- 7: Entrenamiento y validación:**
Inicializar TimeSeriesSplit con F particiones
Para cada $(train_{idx}, val_{idx})$ en $TimeSeriesSplit(S_{train})$:
Entrenar modelo LSTM (θ^k) con $S_{train}[train_{idx}]$
Validar con $S_{train}[val_{idx}]$
Calcular métricas MAE, RMSE, R^2 para entrenamiento y validación
- 8: Evaluación final:**
Entrenar con todo S_{train}
Predecir en S_{test}
Calcular métricas MAE, RMSE, R^2 en prueba
- 9: Comparación:**
Comparar métricas de \hat{S}_{test}^k contra las métricas obtenidas con S_{test} (serie original sin filtrado).
- 10: Retornar resultados:**
Conjunto de métricas {MAE, RMSE, R^2 } para cada combinación filtrada y para la serie original.

Figura 6. Procedimiento general Filtrado Wavelet-LSTM

Obtención de datos

En esta etapa se obtuvieron los datos de velocidad de viento que servirán como base para el desarrollo del modelo. Para este trabajo se utilizó un solo conjunto de datos, que consiste en datos velocidad de viento de Nizanda, Oaxaca, México (Muñoz, 2024). Los datos de velocidad de viento se obtuvieron durante el periodo de 2006-2024 en unidades de metros sobre segundo (m/s). La estación climática está localizada en las coordenadas geográficas 16° 39' 39.6'' N, 95° 0' 26.94'' W, como se muestra en la Figura 5.

Preprocesamiento de datos

El preprocesamiento utilizado incluye limpieza de datos, transformación e integración, las cuales son descritas a continuación.

Para la limpieza se realizó lo siguiente:

- Eliminación de duplicados: Se eliminaron los registros duplicados o que no tenían valores en la variable de interés.
- Selección de características: Se descartaron algunas características que no son necesarias para nuestro experimento. Estas incluyen 'Segment', 'Source', 'Minute', 'ISSRecept', y 'Arc. Int.', estas columnas describen aspectos de los dispositivos de medición. Además, la columna 'Minute' se descartó debido a que los modelos de pronóstico de energía en este estudio utilizan intervalos de una hora.

Dentro del conjunto de datos existen años que no cuentan con todos los días o meses registrados. Ante esta situación, se decidió complementar dichos vacíos con registros de otros años completos, formando combinaciones de dos o más años, teniendo un total de 35040 registros.

El conjunto de datos contiene dos características de velocidad de viento, Wind Speed m/s y High Wind Speed m/s, después de analizar estas características como se muestra en el Cuadro 1 y en los histogramas de la Figura 7, se decidió por utilizar solo la variable High Wind Speed m/s debido a que tiene un rango de

valores más amplio que va de 0 a 23 y que de cierta forma se ajusta más a una distribución normal.

Cuadro 1. Estadísticas descriptivas para las variables de viento del conjunto de datos.

	Wind Speed m/s	High Wind Speed m/s
Media	2.14	11.50
D. E.	1.41	6.92
Min	0	0.00
25%	1.3	5.75
50%	2.2	11.50
75%	3.1	17.25
Max	11.65	23.00

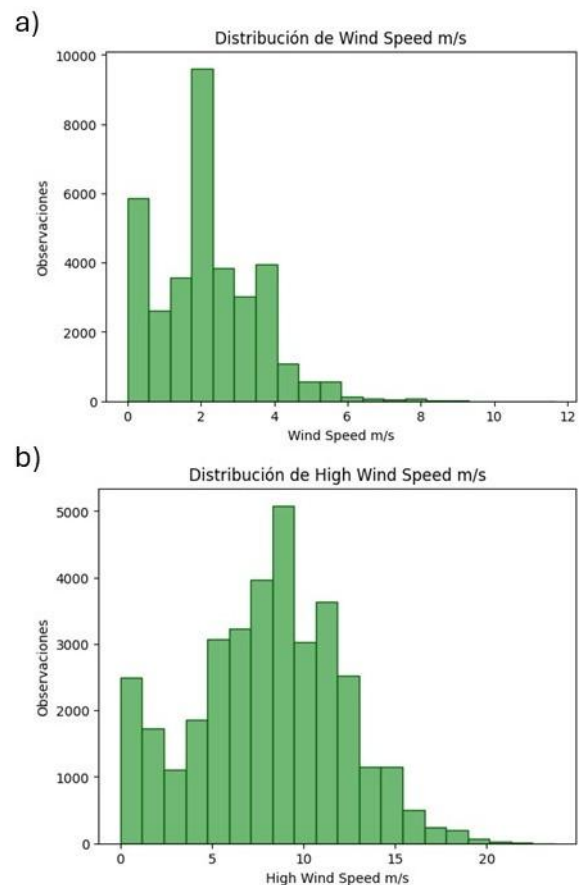


Figura 7. Histograma de las observaciones de velocidad de viento.

Posteriormente, se definieron las secuencias de entrada y salida para el entrenamiento, estableciendo una ventana temporal de 24 h para predecir un horizonte de 1 h. El conjunto de datos se dividió cronológicamente, asignando el 80 % para entrenamiento y el 20 % para prueba. Finalmente, se utilizó MinMaxScaler de scikit-learn para normalizar

los datos y se crearon las secuencias correspondientes para los conjuntos de entrenamiento y prueba.

Transformación wavelet

El procesamiento de la señal se realizó utilizando la librería PyWavelets (Lee *et al.*, 2019). Se aplicó la Transformada Wavelet Discreta (DWT) mediante la función `wavedec` con el modo de periodización, empleando la familia Daubechies 4 (db4) con un nivel de profundidad de 3. Se seleccionó esta wavelet debido a la evidencia reportada por diversos autores como K U and Kovoov (2021); Kio *et al.* (2024); Nascimento *et al.* (2023); Shah *et al.* (2023) y Domínguez-Navarro *et al.* (2021) quienes han demostrado que la familia Daubechies, y en particular la db4, ofrece un buen desempeño en la descomposición de series temporales. Esto permitió descomponer la serie original en un componente de Aproximación (A_3) y tres niveles de Detalles (D_1, D_2, D_3).

Para generar las distintas configuraciones experimentales, se realizó una reconstrucción parcial mediante transformada inversa con la función `waverec`. Este procedimiento consistió en combinar el componente de aproximación con niveles de detalle específicos, asignando valores de cero a los coeficientes restantes que no eran de interés para cada modelo evaluado. Explicado formalmente en las Ecuaciones de la 9 a la 13.

Sea la serie de tiempo:

$$S = \{s_1, s_2, \dots, s_n\}, \quad s_t \in \mathbb{R} \quad (9)$$

Aplicamos la DWT hasta nivel L:

$$W(S) = \{A_L, D_1, D_2, \dots, D_L\} \quad (10)$$

Donde A_L es el coeficiente de aproximación en el nivel L y D_i son los coeficientes de detalle en cada nivel i.

Para cada subconjunto de detalle $D^{(k)} \subseteq \{D_1, D_2, \dots, D_L\}$, definimos la combinación de coeficientes:

$$C^{(k)} = A_L + \sum_{D_i \in D^{(k)}} D_i \quad (11)$$

La serie filtrada correspondiente a la combinación $C^{(k)}$ se obtiene aplicando la transformada inversa:

$$\hat{S}^{(k)} = W^{-1}(C^{(k)}) \quad (12)$$

El conjunto de todas las series filtradas es:

$$S_{filtradas} = \{\hat{S}^{(1)}, \hat{S}^{(2)}, \dots, \hat{S}^{(K)}\} \quad (13)$$

En la Figura 9 se muestran los primeros cien registros de la serie original (a) y las comparaciones de las series filtradas obtenidas respecto a la original (b-h) en la cual se observa cómo afecta la reconstrucción parcial a los datos originales.

Entrenamiento del modelo

La arquitectura de la red neuronal LSTM se optimizó mediante un ajuste de hiperparámetros utilizando el método Hyperband (Li *et al.*, 2018). Para cada serie filtrada obtenida mediante la reconstrucción wavelet, se llevó a cabo un proceso independiente de ajuste. De este modo, cada combinación de componentes de aproximación y detalle cuenta con su propia búsqueda de hiperparámetros. Además, durante el entrenamiento se aplicó una estrategia de validación cruzada específica para series de tiempo, con el fin de evaluar de manera más robusta el desempeño de cada configuración y reducir el riesgo de sobreajuste (Bergmeir & Benítez, 2012). El algoritmo Hyperband (Figura 8) asigna inicialmente pocos recursos (épocas de entrenamiento) a un gran número de configuraciones y, a medida que avanza, descarta de forma temprana aquellas con desempeño deficiente, concentrando los recursos en las mejores. El criterio de parada se basa en la métrica raíz del error cuadrático medio (RMSE). El espacio de búsqueda definido para optimización se detalla en el Cuadro 2. Estos hiperparámetros corresponden a los parámetros de arquitectura y optimización disponibles en la implementación oficial de PyTorch (PyTorch, 2026). En el Cuadro 3 se muestran los hiperparámetros obtenidos para el modelo base y el mejor modelo.

Entrada: R, η

Iniciación: $s_{\max} = \lceil \log_{\eta}(R) \rceil, B = (s_{\max} + 1)$

Para $s \in \{s_{\max}, s_{\max} - 1, \dots, \}$ **hacer**

$$n = \left\lceil \frac{B}{R} \cdot \frac{\eta^s}{s+1} \right\rceil, r = R \cdot \eta^{-s}$$

/*

comienza SuccessiveHalving con el bucle interno

*/

$T = \text{obtener_configuración_hiperparametro}(n)$

Para $i \in \{0, \dots, s\}$ **hacer**

$$n_i = \lfloor n \cdot \eta^{-i} \rfloor$$

$$r_i = r \cdot \eta^i$$

$$L = \{\text{calcular_metrica}(t, r_i) : t \in T\}$$

$$T = \text{top_k}(T, L, \lfloor n_i/\eta \rfloor)$$

Final

Final

Devuelve Configuración con menor error observada hasta el momento.

Figura 8. Algoritmo Hyperband basado en (Li *et al.*, 2018)

Evaluación

El desempeño de los modelos propuestos se evaluó comparando las predicciones con los valores reales del conjunto de prueba. Se utilizaron las siguientes métricas de error: Error Absoluto Medio (MAE), Raíz del Error Cuadrático Medio (RMSE) y el Coeficiente de Determinación (R^2).

Las Ecuaciones 14 a 16 describen el cálculo de estas métricas, donde y_i representa el valor real observado, \hat{y}_i el valor pronosticado por el modelo, \bar{y} el promedio de los valores reales y n el número total de muestras evaluadas.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (14)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (15)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (16)$$

Cuadro 2. Espacio de búsqueda para ajuste de hiperparámetros

Hiperparámetros	Valores
hidden_size	16, 32, 48, 64, 80
num_layers	1, 3
dropout	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
learning_rate	[1e-4, 1e-2]
weight_decay	[1e-6, 1e-2]
batch_size	16, 32, 64
optimizer	"adam", "rmsprop"
bidireccional	False, True

Cuadro 3. Hiperparámetros para el modelo base y el mejor modelo del enfoque de filtrado wavelet.

Hiperparámetros	Base	Aprox + D1, D2
hidden_size	48	16
num_layers	3	1
dropout	0	0
learning_rate	0.001337548	0.002043936
weight_decay	3.37E-05	2.50E-06
batch_size	32	64
optimizer	adam	adam
bidireccional	True	True

El MAE mide la magnitud promedio de los errores en un conjunto de predicciones, sin considerar su dirección. El RMSE es una regla de puntuación cuadrática que también mide la magnitud promedio del error, pero penaliza más severamente los errores grandes. El coeficiente R^2 proporciona una medida de qué tan bien se replican los resultados observados por el modelo, basándose en la proporción de la variación total de los resultados explicada por el modelo (Chicco *et al.*, 2021).

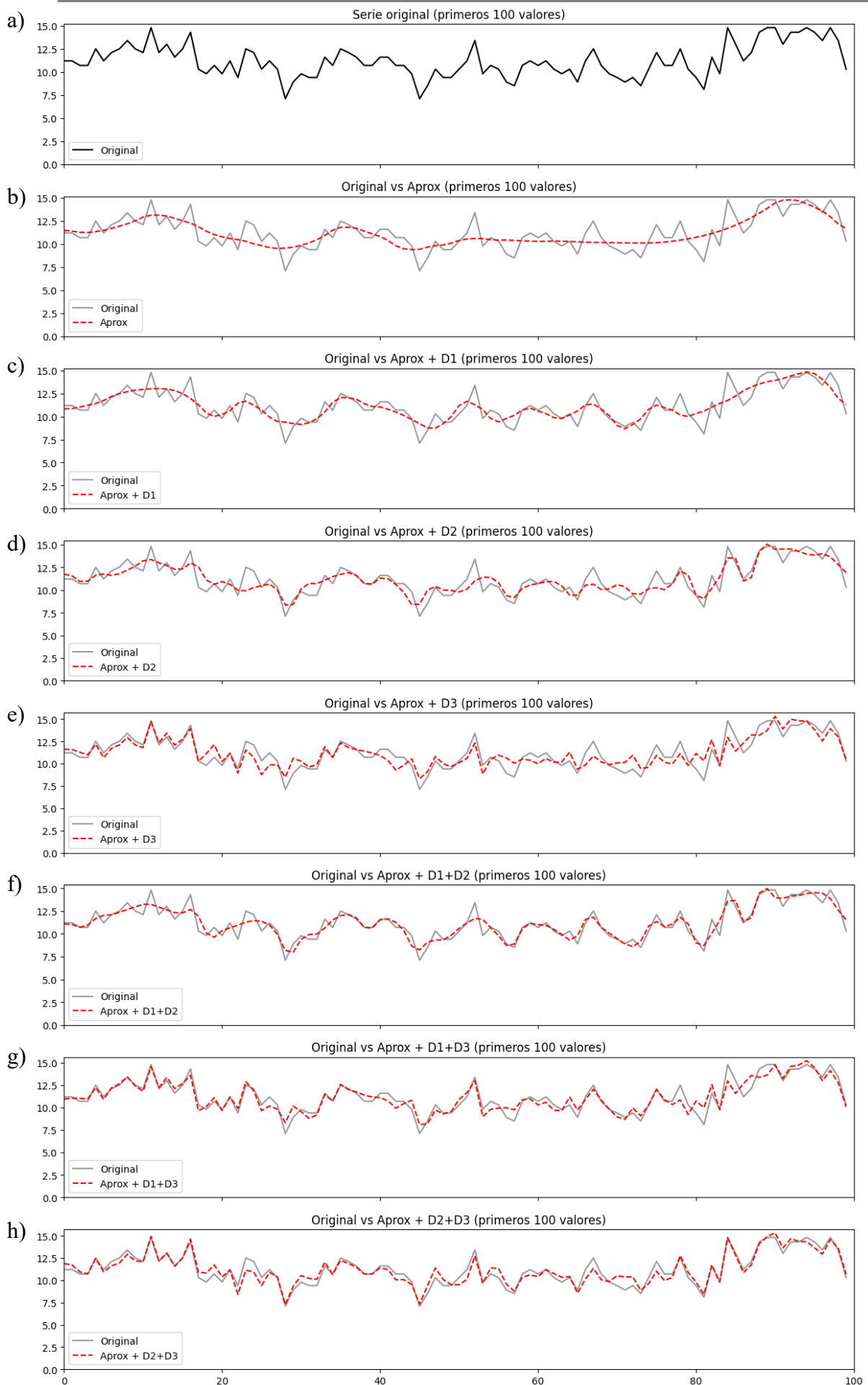


Figura 9. Grafica de serie original y reconstrucción parcial con transformada wavelet discreta

RESULTADOS Y DISCUSIÓN

Para evaluar el impacto de la transformada wavelet con reconstrucción parcial en el pronóstico de series de tiempo, se comparó el desempeño de un modelo LSTM base (sin transformación wavelet) además de modelos de referencia *AutoRegressive Integrated Moving Average* (ARIMA), *Seasonal ARIMA* (SARIMA), Persistencia, Random Forest (RF), Support Vector Regression (SVR) y Xtreme Gradient Boosting (XGBoost) respecto a modelos entrenados con series reconstruidas mediante diferentes combinaciones de los coeficientes de Aproximación (Aprox) y Detalle (D_n) utilizando de igual forma modelos LSTM.

El rendimiento se midió utilizando las métricas antes mencionadas. El Cuadro 4 muestra el resumen de las métricas obtenidas durante las fases de entrenamiento y prueba (marcadas en gris y blanco respectivamente), donde Aprox se refiere al componente de Aproximación y D_i se refiere a los componentes de detalle. Los mejores resultados aparecen resaltados en negritas. Asimismo, la Figura 10 muestra una gráfica comparativa de los resultados del coeficiente de determinación R^2 para el modelo base y los modelos procesados de las reconstrucciones parciales con transformada wavelet discreta.

Cuadro 4. Resultados de entrenamiento y prueba

Modelo	MAE	RMSE	R^2
Base	1.1710	1.5402	0.8623
	1.2716	1.6808	0.8059
Aprox	1.0500	1.3510	0.8940
	1.1590	1.4870	0.8480
Aprox + D1	0.8190	1.0680	0.9340
	0.9070	1.1900	0.9030
Aprox + D2	0.9400	1.2130	0.9150
	1.0190	1.3120	0.8820
Aprox + D3	1.1200	1.4460	0.8790
	1.2170	1.5740	0.8300
Aprox + D1, D2	0.6690	0.8830	0.9550
	0.7420	0.9790	0.9340
Aprox + D1, D3	1.1110	1.4320	0.8810
	1.2170	1.5790	0.8290
Aprox + D2, D3	1.2390	1.6000	0.8510
	1.3230	1.7210	0.7960

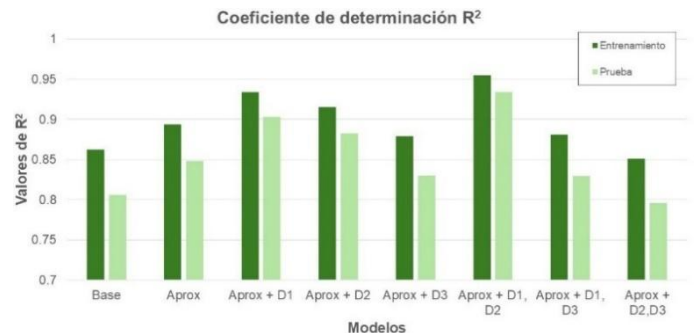


Figura 10. Coeficiente de determinación R^2 para conjuntos de entrenamiento y prueba.

Con el propósito de ilustrar el desempeño del modelo más destacado obtenido mediante la estrategia de filtrado con la Transformada Wavelet Discreta, en la Figura 11 se presenta un segmento de prueba de los registros de velocidad del viento (en color azul), junto con los resultados de predicción del modelo base-LSTM (en color verde) y del modelo wavelet-LSTM (en color naranja). Este último corresponde al modelo entrenado con la reconstrucción parcial que integra la componente de aproximación y los detalles uno y dos.

En el Cuadro 5 se muestra la comparación del rendimiento de modelos clásicos de series de tiempo (ARIMA, SARIMA y Persistencia) y modelos usados comúnmente en aprendizaje automático (RF, SVR y XGBoost) con el mejor modelo obtenido del enfoque de filtrado con transformada wavelet discreta.

Con base en los datos presentados anteriormente, se observa que la estrategia de filtrado con reconstrucción parcial wavelet supera tanto al modelo base como a los modelos clásicos de series de tiempo y modelos comúnmente utilizados en aprendizaje automático, en específico para la configuración Aprox + D1, D2 con la cual se obtuvo el mejor resultado alcanzando un R^2 de 0.9340 y reduciendo el RMSE a 0.9790 en comparación con el modelo base el cual tiene un R^2 de 0.8059 y un RMSE de 1.6808, esto para el segmento de prueba. Además, se observa que los modelos mejoran al agregar el componente D1,

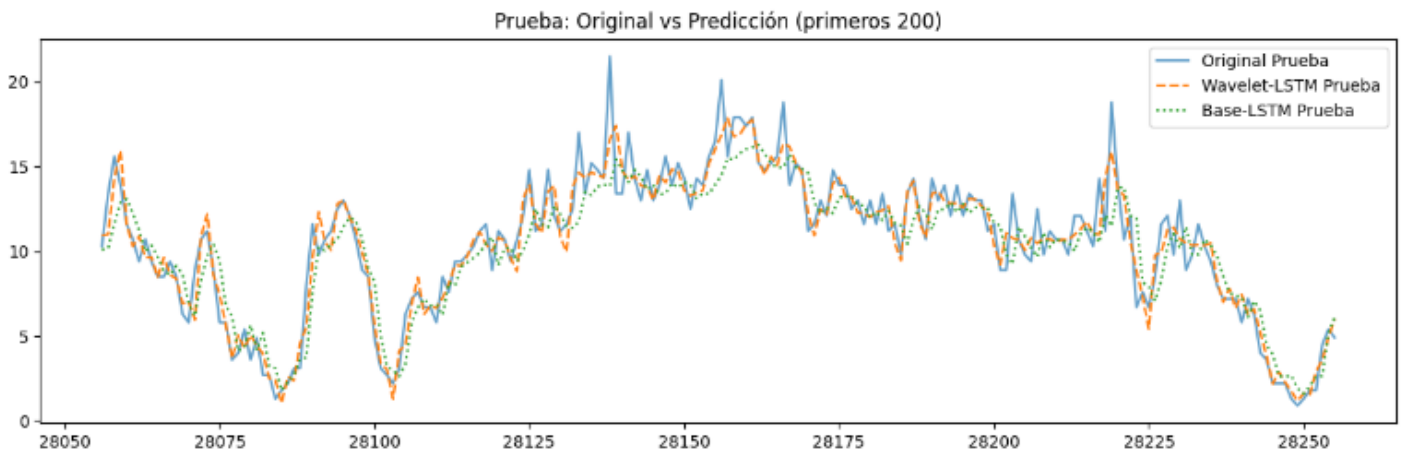


Figura 11. Comparación de resultados del modelo base, Aprox + D1, D2 y los datos originales

Cuadro 5. Comparación de los resultados con modelos de referencia.

Modelo	MAE	RMSE	R ²
Base	1.1710	1.5402	0.8623
	1.2716	1.6808	0.8059
ARIMA	1.1844	1.5779	0.8555
	3.6909	4.4709	-0.3742
SARIMA	1.1827	1.5729	0.8564
	3.5674	4.3439	-0.2972
Persistencia	1.2008	1.6305	0.8457
	1.3669	1.8216	0.7719
RF	1.0302	1.3360	0.8964
	1.2799	1.6821	0.8055
SVR	1.0638	1.4706	0.8745
	1.2875	1.7124	0.7984
XGBoost	1.0339	1.3566	0.8932
	1.2755	1.6763	0.8068
Aprox + D1, D2	0.6690	0.8830	0.9550
	0.7420	0.9790	0.9340

esto podría atribuirse a que el componente D1 contiene la información de fluctuaciones de alta frecuencia, la cual al combinarse con el componente Aprox (tendencia general) resulte ser más sencillo de capturar patrones por el modelo LSTM. Por otro lado, se observa que en las configuraciones con el componente D3 bajó su rendimiento incluso por debajo del modelo base en el caso de Aprox + D2, D3.

CONCLUSIONES

En este trabajo se comparó el uso de la transformada wavelet como técnica de preprocesamiento para el pronóstico de series de tiempo a corto plazo, utilizando redes neuronales (LSTM). Los resultados demostraron que el uso de la transformada wavelet como filtro multi-resolución permite superar las limitaciones de los modelos clásicos de series de tiempo, los modelos comunes de aprendizaje automático y el modelo LSTM convencional.

La mejor configuración se obtuvo mediante la combinación de la aproximación y los dos primeros niveles de detalle (Aprox + D1, D2). Este resultado podría explicarse porque dicha reconstrucción con los componentes de aproximación y detalles 1 y 2, atenúa la variabilidad de la serie de velocidad de viento, lo que facilita al modelo LSTM capturar patrones de manera más efectiva.

Como trabajo futuro, se propone implementar algoritmos de optimización metaheurísticos para la selección de componentes de descomposición. Asimismo, se sugiere extender la evaluación de esta metodología hacia horizontes de predicción más amplios.

AGRADECIMIENTOS

Brandon Nahú Nava Martínez agradece a la Secretaría de Ciencia, Humanidades, Tecnología e

Innovación (SECIHTI), por el apoyo financiero brindado para la realización de sus estudios de posgrado. Este trabajo fue parcialmente financiado por el Programa Presupuestario F003 Ciencia Básica y de Frontera del Secihti, a través del proyecto con número de subvención CBF-2025-G-1233.

CONFLICTO DE INTERESES

Los autores declaran no tener ningún conflicto de intereses.

ORCID DE LOS AUTORES

<https://orcid.org/0009-0009-3386-5506>
<https://orcid.org/0000-0002-1276-6957>
<https://orcid.org/0000-0002-6481-4316>
<https://orcid.org/0000-0003-1834-2431>
<https://orcid.org/0000-0002-0332-5658>
<https://orcid.org/0000-0002-0794-8967>

REFERENCIAS

- Amirteimoury, F., Keynia, F., Amirteimoury, E., Memarzadeh, G., & Shabaniyan, H. (2025). A novel wind speed prediction model based on neural networks, wavelet transformation, mutual information, and coot optimization algorithm. *Sci Rep*, *15*(1), 10860. <https://doi.org/10.1038/s41598-025-94082-2>
- Azamar Alonso, A., & García Beltrán, Y. M. (2021). Diagnóstico y riesgos de la energía eólica en México. *Revista de Geografía Agrícola*(67), 27-45. <https://doi.org/10.5154/r.rga.2021.67.02>
- Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192-213. <https://doi.org/10.1016/j.ins.2011.12.028>
- Buratto, W. G., Muniz, R. N., Nied, A., Barros, C. F. d. O., Cardoso, R., & Gonzalez, G. V. (2024). Wavelet CNN-LSTM time series forecasting of electricity power generation considering biomass thermal systems. *IET Generation, Transmission & Distribution*, *18*(21), 3437-3451. <https://doi.org/10.1049/gtd2.13292>
- Cano Torres, L. R., & Rodríguez Cruz, L. A. (2020). El impacto social de las energías limpias en comunidades vulnerables. La energía eólica en la comunidad zapoteca de Juchitán de Zaragoza, Oaxaca. *Ambiente y Desarrollo*, *24*(46), 1-18. <https://doi.org/10.11144/Javeriana.ayd24-46.isel>
- Castro, L. R., & Castro, S. M. (1995). *Wavelets y sus aplicaciones I Congreso Argentino de Ciencias de la Computación.*
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput Sci*, *7*, e623. <https://doi.org/10.7717/peerj-cs.623>
- Daubechies, I. (1992). *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9781611970104>
- Domínguez-Navarro, J. A., Lopez-Garcia, T. B., & Valdivia-Bautista, S. M. (2021). Applying Wavelet Filters in Wind Forecasting Methods. *Energies*, *14*(11). <https://doi.org/10.3390/en14113181>
- Eólica, A. M. d. E. (2015). *El potencial eólico mexicano: Oportunidades y retos en el nuevo sector eléctrico.*
- Guo, X., Zhu, C., Hao, J., Kong, L., & Zhang, S. (2023). A Point-Interval Forecasting Method for Wind Speed Using Improved Wild Horse Optimization Algorithm and Ensemble Learning. *Sustainability*, *16*(1). <https://doi.org/10.3390/su16010094>
- K U, J., & Kovoov, B. C. (2021). A Wavelet-based hybrid multi-step Wind Speed Forecasting model using LSTM and SVR. *Wind Engineering*, *45*(5), 1123-1144. <https://doi.org/10.1177/0309524x20964762>
- Kio, A. E., Xu, J., Gautam, N., & Ding, Y. (2024). Wavelet decomposition and neural networks: a potent combination for short term wind speed and power forecasting. *Frontiers in Energy Research*, *12*. <https://doi.org/10.3389/fenrg.2024.1277464>
- Lee, G. R., Gommers, R., Wasilewski, F., Wohlfahrt, K., & O'Leary, A. (2019). PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, *4*(36), 1237. <https://doi.org/10.21105/joss.01237>
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, *18*(185), 1-52.
- Li, W., & Law, K. L. E. (2024). Deep Learning Models for Time Series Forecasting: A Review. *IEEE Access*, *12*, 92306-92327. <https://doi.org/10.1109/access.2024.3422528>
- Muñoz, R. B., F.; Lebrija-Trejos, E.; Gallardo-Cruz, J. A.; Enríquez, M.; Romero-Romero, M. A.; López-Mendoza, R. D.; Meave, J. A. (2024). *Daily weather data from Nizanda, Mexico (2006–2024)*. <https://doi.org/10.5281/zenodo.7970409>

- Nascimento, E. G. S., de Melo, T. A. C., & Moreira, D. M. (2023). A transformer-based deep neural network with wavelet transform for forecasting wind speed and wind energy. *Energy*, 278. <https://doi.org/10.1016/j.energy.2023.127678>
- PyTorch. (2026). *torch.nn.LSTM* — *PyTorch documentation*. <https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html>
- Shah, A. A., Aftab, A. A., Han, X., Baloch, M. H., Honnurvali, M. S., & Chauhdary, S. T. (2023). Prediction Error-Based Power Forecasting of Wind Energy System Using Hybrid WT–ROPSO–NARMAX Model. *Energies*, 16(7). <https://doi.org/10.3390/en16073295>
- Soman, S. S., Zareipour, H., Malik, O., & Mandal, P. (2010). A Review of Wind Power and Wind Speed Forecasting Methods With Different Time Horizons.pdf. *North American power symposium*, 8. <https://doi.org/10.1109/NAPS.2010.5619586>
- Tamilselvi, C., Paul, R. K., Yeasin, M., & Paul, A. K. (2024). Novel wavelet-LSTM approach for time series prediction. *Neural Computing and Applications*, 37(17), 10521-10530. <https://doi.org/10.1007/s00521-024-10561-z>
- Vishnutheerth, E. P., Vijay, V., Satheesh, R., & Kolhe, M. L. (2024). A Comprehensive Approach to Wind Power Forecasting Using Advanced Hybrid Neural Networks. *IEEE Access*, 12, 124790-124800. <https://doi.org/10.1109/access.2024.3450096>
- Wang, Y., Zou, R., Liu, F., Zhang, L., & Liu, Q. (2021). A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy*, 304, 117766. <https://doi.org/https://doi.org/10.1016/j.apenergy.2021.117766>
- Yu, B., Lu, Z., & Qian, W. (2025). Wavelet-denoised graph-Informer for accurate and stable wind speed prediction. *Applied Soft Computing*, 176, 113182. <https://doi.org/https://doi.org/10.1016/j.asoc.2025.113182>