

Pronóstico de la velocidad del viento mediante técnicas de Inteligencia Artificial

Juan Ramón V. Barraza¹, Hector Rodriguez R.¹, Víctor González Huitrón¹,
Carlos A. Lara Alvarez³, Juan J. Flores²

¹ Instituto Tecnológico de Culiacán, División de Estudios de Posgrado e Investigación, Culiacán, Sinaloa, México

² Universidad Michoacana de San Nicolás de Hidalgo, Facultad de Ingeniería Eléctrica, División de Estudios de Posgrado, Morelia, Michoacán, México

³ Centro de Investigación en Matemáticas, A.C. Unidad Zacatecas, Zacatecas, México
juan.valenzuela@itculiacan.edu.mx, hrodriguez@itculiacan.edu.mx,
victor.gonzalez@conacyt.mx, carlos.lara@cimat.mx, juanf@umich.mx

Resumen. Realizar pronósticos de múltiples pasos en el tiempo no es una tarea sencilla; por lo tanto, es importante desarrollar y usar técnicas que simplifiquen esta tarea. Este trabajo prueba modelos de Inteligencia Artificial para tres diferentes estrategias de pronósticos a múltiples pasos: *Iterativa*, *Directa*, y *Multiple Input - Multiple Output (MIMO)*. El primer paso consiste en preprocesar las series de tiempo para *limpiar* (eliminar datos atípicos y ruido) y *transformar* los datos en secuencias o imágenes; después, se usan tres tipos de *Artificial Neural Networks (ANN)* para construir modelos de pronóstico: *Multilayer Perceptron (MLP)*, *Long Short Term Memory (LSTM)* y *Convolutional Neural Networks (CNN)*. Se muestran experimentos con series de tiempo obtenidas en estaciones climáticas en Michoacán, México, donde se puede observar el desempeño que se obtuvo con cada modelo y estrategia.

Palabras clave: inteligencia artificial, pronóstico, velocidad del viento.

Wind Speed Forecast Using Artificial Intelligence Techniques

Abstract. *N*-steps forecasting is not an easy task; it is important to develop and use techniques that simplify this task. This paper tests Artificial Intelligence models for three different multi-step forecasting strategies: *Iterative*, *Direct*, and *Multiple Input - Multiple Output (MIMO)*. The first step preprocesses the time-series to *clean* (remove outliers and noise) and *transform* the data into sequences or images; after that, three types of Artificial Neural Networks (ANN) are used to construct forecasting models: *Multilayer Perceptron (MLP)*, *Long Short Term Memory (LSTM)* and *Convolutional Neural Networks (CNN)*. Experiments that use data generated from climate stations in Michoacan, Mexico are shown, where the performance obtained with each model and strategy can be observed.

Keywords: artificial intelligence, forecasting, wind speed.

1. Introducción

En la actualidad la energía eléctrica es la base para muchas de las actividades realizadas por la humanidad; sin embargo, su producción puede utilizar recursos no renovables y deteriorar el medio ambiente. Durante el año 2017 la generación de energías limpias en México fue del 21.08 % del cual solo el 3.23 % es energía eólica [1].

Aunque la energía eólica tiene muchas ventajas, la mayor dificultad es que la velocidad del viento no es constante a través del tiempo. Por este motivo, es muy importante pronosticar la velocidad del viento —y en consecuencia, conocer la cantidad de energía limpia que se puede producir. Sin embargo, la tarea de pronosticar la velocidad del viento no es una tarea sencilla, especialmente cuando se desea pronosticar un horizonte amplio.

Los modelos de pronóstico que usan Inteligencia Artificial se han popularizado en los últimos años porque tienen mejor desempeño que los métodos estadísticos clásicos [2]. El diseño de estos modelos es importante porque afecta drásticamente su velocidad y eficiencia [4].

El objetivo de este trabajo es comparar y presentar las fortalezas y desventajas de tres tipos de *Artificial Neural Networks* (ANN) —*Multilayer Perceptron*, *Long Short Term Memory* (LSTM) y *Convolutional Neural Network* (CNN)— para pronóstico directo, recursivo y MIMO. Además se explica el preprocesamiento (limpieza y transformación) de series de tiempo para su uso en modelos de pronóstico. En especial, se describe la técnica *Gramian Angular Field* que se usa para transformar series de tiempo a imágenes, lo que permite analizar las series con una CNN.

El resto de este trabajo se estructura de la siguiente forma: la Sección 2, presenta trabajos relacionados e implementaciones similares, la Sección 3 explica los modelos de IA utilizados, la Sección 4 detalla la metodología utilizada, la Sección 5 muestra los resultados obtenidos y Sección 6 concluye este trabajo.

2. Trabajos relacionados

El pronóstico de series de tiempo es de creciente interés en muchas áreas de la ciencia e ingeniería —v.g., economía, finanzas, meteorología y telecomunicaciones. El pronóstico de viento a múltiples pasos consiste en pronosticar un rango más amplio de tiempo con intervalo definido. Los problemas de pronóstico a múltiples pasos se pueden clasificar en: super-corto, corto, mediano, y largo plazo según sea su horizonte de tiempo [2]. A diferencia de pronosticar el paso siguiente en el tiempo, los pronósticos con un horizonte más amplio son más difíciles porque el error y la incertidumbre se acumulan, lo que reduce la precisión [4].

Además, la estrategia que use un pronosticador es de vital importancia, ya que afecta el costo computacional y precisión del pronosticador.

Los modelos de aprendizaje de máquina se han popularizado en las últimas décadas como serios contendientes de los modelos de pronóstico estadísticos clásicos [5]. Entre los modelos de aprendizaje de máquina se encuentran los modelos de aprendizaje profundo, como lo son las *Artificial Neural Networks: Multilayer Perceptron* (MLP), *Long Short Term Memory* (LSTM) y *Convolutional Neural Network* (CNN).

Pasero & Ruffa [6] implementan un enfoque de pronóstico a largo plazo con un modelo MLP y una estrategia recursiva; sin embargo, sus resultados muestran que un modelo de *Support Vector Machine* (SVM) con estrategia directa consigue mejores resultados. Por otro lado, las LSTM tienen la característica de poder procesar grandes cantidades de datos y poseen la capacidad de generalización; es decir, se adaptan a datos no conocidos. Con mejores resultados que los modelos basados en SVM [7].

Las CNN son redes ampliamente conocidas por su uso en clasificación de imágenes. Estas al igual que las LSTM tienen la capacidad de procesar grandes cantidades de información, Anwen, et al. [8] utiliza CNN para realizar pronósticos de viento para múltiples pasos.

Los pronósticos a múltiples pasos que se obtienen con la estrategia iterativa tienen la desventaja de deteriorarse al acumular errores, ya que solamente se entrena un modelo a un paso; sin embargo, su resultado es comparable al trabajar un horizonte limitado [9].

Una solución para evitar la deterioración al acumular errores tal como en la estrategia iterativa es entrenar un modelo por cada paso en el horizonte de tiempo como se propone en la estrategia directa [4]. Quin, et al. [10] muestran el uso de la estrategia directa en conjunto para pronósticos de la velocidad del viento a cuatro horas.

Otra alternativa es la estrategia MIMO que se caracteriza por ser un solo modelo que predice múltiples pasos en el tiempo [4]. Li, et al. [11] comparan las estrategias iterativa y MIMO, para obtener el pronóstico de velocidad de viento a corto plazo donde MIMO obtiene mejores resultados.

3. Modelos de Inteligencia Artificial

Esta sección revisa los conceptos fundamentales de algunos modelos de inteligencia artificial que se utilizan en este trabajo para realizar pronósticos de la velocidad del viento.

3.1. Artificial Neural Network

El *Multilayer Perceptron* (MLP) es uno de los tipos de redes más utilizados, el cual típicamente se conforma de múltiples capas y nodos [3]. La figura 1 muestra la arquitectura de un MLP, donde sus entradas son valores históricos, y como salida el pronóstico del viento.

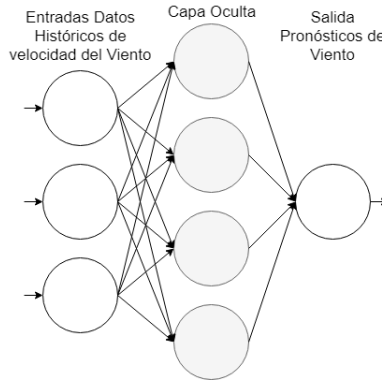


Fig. 1. Diagrama representación de Red Neuronal Artificial MLP

3.2. Long Short Term Memory

Las Redes Neuronales LSTM son muy similares a las MLP en su estructura, tienen capas de entrada, capas ocultas y capa de salida. Sin embargo, LSTM en su capa oculta poseen una unidad de memoria [7].

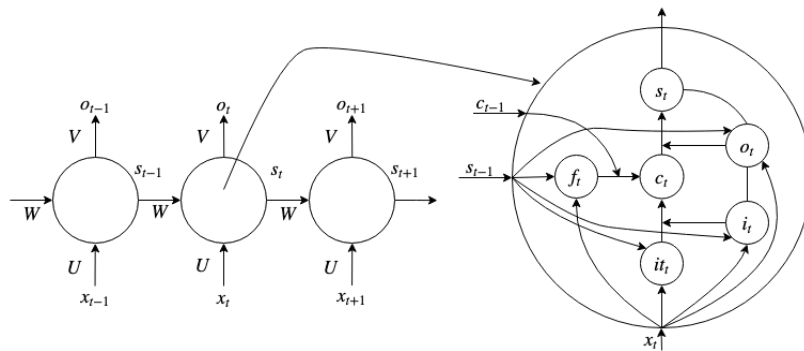


Fig. 2. Diagrama de funcionamiento de Long Short Term Memory

La unidad de memoria consiste de tres compuertas (*Input Gate* (i_t), *Forget Gate* (f_t) y *Output Gate* (o_t)) y una conexión recurrente. La unidad tiene una entrada x_t y dos retroalimentaciones de estados previos de la unidad que son la salida del estado previo s_{t-1} y la variable de estado c_{t-1} . Las compuertas utilizan una función de activación sigmoide g , mientras que los estados utilizan una función \tanh . La unidad de memoria de una LSTM se puede definir por las siguientes expresiones:

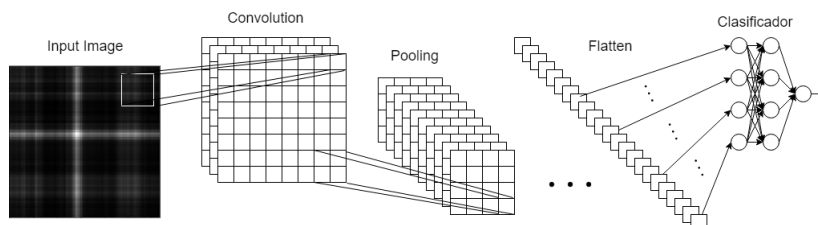


Fig. 3. Diagrama de funcionamiento de una Red Neuronal Convolucionada

$$\begin{aligned}
 \text{Input Gate} & \quad i_t = g(W_{xi}x_t + W_{hi}s_{t-1} + b_i), \\
 \text{Forget Gate} & \quad f_t = g(W_{xf}x_t + W_{hf}s_{t-1} + b_f), \\
 \text{Output Gate} & \quad o_t = g(W_{xo}x_t + W_{ho}s_{t-1} + b_o), \\
 \text{Input Transform} & \quad it_t = \tanh(W_{xc}x_t + W_{hc}s_{t-1} + b_{it_t}), \\
 \text{State Update} & \quad c_t = f_t \cdot c_{t-1} + i_t \cdot it_t, \\
 \text{Output} & \quad s_t = o_t \cdot \tanh(c_t),
 \end{aligned} \tag{1}$$

donde W son parámetros y b es el sesgo [7].

3.3. Redes neuronales convolucionales

Las Redes Neuronales Convolucionales son capaces de extraer características de imágenes de dos dimensiones [8]. Estas redes tienen cuatro etapas principales: Convolución, agrupación (pooling), aplanado de datos (flatten) y clasificador. En la figura 3 se muestra el proceso generalizado que realiza una CNN.

La etapa de *Convolución* consiste en extraer características a partir de recorridos en la imagen, mientras que el *pooling* realiza un sub-muestreo para reducir el costo computacional requerido al disminuir las dimensiones y parámetros previamente extraídos durante una convolución. Posteriormente se tiene la etapa de *flatten* que aplanan los datos a un vector para finalmente alimentar las características extraídas a un clasificador.

4. Metodología

La figura 4 muestra el proceso general utilizado en este trabajo, la entrada es una serie de tiempo y la salida es un modelo de predicción del viento para n pasos futuros. El proceso consiste de cuatro etapas principales: (i) selección de estrategia, (ii) selección de modelo de IA, (iii) preprocesamiento de Datos, y (iv) entrenamiento y ajuste de modelo.

4.1. Estrategias de predicción múltiples pasos en el tiempo

A diferencia del pronóstico del siguiente valor de una serie de tiempo, el pronóstico de múltiples pasos en el tiempo es considerablemente más difícil

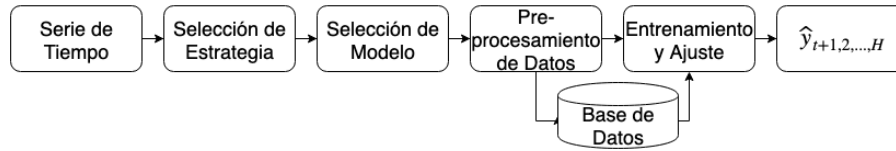


Fig. 4. Diagrama del proceso general de Serie de Tiempo a modelo predictivo con n pasos a futuro

debido a la acumulación de errores, reducción de precisión, e incremento de incertidumbre [3]. Por este motivo, se han propuesto diferentes estrategias para mejorar la aptitud de los pronósticos. Taieb et al. [4] hacen una revisión de las técnicas conocidas. En este trabajo se utilizan tres estrategias: Iterativa, Directa y MIMO.

Estrategia Iterativa. La Estrategia Iterativa consiste en entrenar un modelo f para un solo paso en el tiempo [4]; es decir,

$$y_{t+1} = f(y_t, \dots, y_{t-d+1}) + r, \quad (2)$$

donde $t \in \{d, \dots, N - 1\}$. Cuando se busca pronosticar H pasos en el futuro, primero se pronostica un paso aplicando el modelo f . El valor obtenido en el primer pronóstico se alimenta al modelo para pronosticar el siguiente paso en el tiempo (figura 5), repitiendo el proceso hasta predecir el horizonte H completo [3].

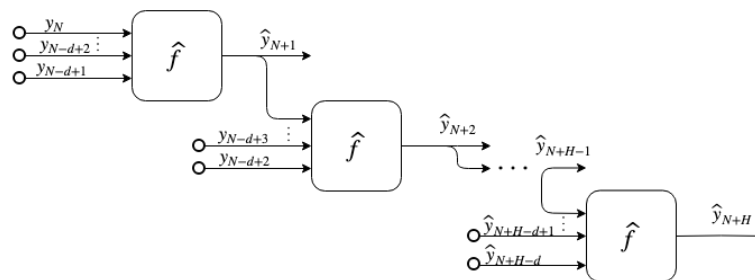


Fig. 5. Estrategia Iterativa para Pronóstico de Series de Tiempo para H pasos a futuro.

En (3) se observa que \hat{y} se estima a partir de entradas de la serie y valores estimados. Cuando $h = 1$, el modelo solo usa valores de la serie para pronosticar un solo paso. Si $2 \geq h \geq d$, el modelo se alimenta de valores estimados y de la serie de tiempo; finalmente, cuando $h > d$, el modelo se alimenta solo de valores pronosticados.

Algoritmo 1 Algoritmo para Estrategia Iterativa

```

1: procedure ITERATIVE_STRATEGY(model, data, horizon, window_length)
2:   queue = data
3:   predictions = []
4:   for h = 0; 0 ≤ horizon; h = h + 1 do
5:     current = queue[-window_length :]
6:     predicted = model.predict(current)
7:     queue = queue.append(predicted)
8:     predictions.add(predicted)
9:   end for
   return predictions
10: end procedure

```

$$\hat{y}_{N+h} = \begin{cases} \text{If } h = 1 & \hat{f}(y_N, \dots, y_{N-d}), \\ \text{If } h \in (2, \dots, d) & \hat{f}(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+1}, \dots, y_{N-d+h}), \\ \text{If } h \in (d+1, \dots, H) & \hat{f}(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+h-d}). \end{cases} \quad (3)$$

La estrategia iterativa sufre de la acumulación de errores por cada paso que recorre para $h > 1$ hasta llegar al horizonte máximo H . Esto es especialmente cierto cuando $h > d$, como plantea el tercer caso de (3), donde el modelo comienza a ser alimentado con solamente valores previamente pronosticados [4].

El Alg. 1 muestra la secuencia lógica utilizada en este trabajo para realizar pronósticos con la estrategia iterativa. El algoritmo recibe el modelo (*model*), los datos de entrada (*data*), el horizonte H (*horizon*) y el tamaño de ventana de datos (*window_length*). El proceso consiste en una cola (*queue*) iniciando con los datos (*data*) como punto inicial. Posteriormente, se inicia un ciclo For desde $h = 0$ hasta el horizonte H , avanzando un solo paso por ciclo. Finalmente, se realiza pronóstico con datos en cola y se actualiza.

Estrategia Directa. Se caracteriza por pronosticar cada paso del horizonte en tiempo de manera independiente [4]. Es decir, si se tiene un horizonte H , se debe entrenar la misma cantidad de modelos para cada paso de tiempo que conforma el horizonte completo. Sea su representación dada por:

$$y_{t+h} = f_h(y_t, \dots, y_{t-d+1}) + r, \quad (4)$$

donde $t \in (d, \dots, N - H)$, $h \in (1, \dots, H)$ y la constante r representa el ruido externo que pueda tener la serie de tiempo.

La Estrategia Directa (figura 6) presenta la ventaja de no utilizar valores aproximados ya que no retroalimenta datos anteriormente pronosticados, además de poseer la capacidad de ajustar cada modelo pronosticador individualmente.

En Alg. 2 se puede observar la lógica de programación utilizada para realizar pronósticos con la estrategia directa. La lógica utilizada es muy similar a la

Algoritmo 2 Algoritmo para Estrategia Directa

```

1: procedure DIRECT_STRATEGY(models, data, horizon)
2:   predictions = []
3:   for  $h = 0; 0 \leq horizon; h = h + 1$  do
4:     model = models[h]
5:     predicted = model.predict(data)
6:     predictions.add(predicted)
7:   end for
   return predictions
8: end procedure

```

utilizada por la estrategia iterativa, siendo la mayor diferencia un recorrido realizado a una colección que contiene los modelos para realizar cada pronóstico.

A pesar de las ventajas que presenta la estrategia directa, el costo computacional requerido para el entrenamiento individual y el ajuste de los parámetros de cada modelo es una gran desventaja, por lo que se debe considerar al comparar con otras estrategias.

Estrategia MIMO. La Estrategia MIMO (Multiple Input - Multiple Output) posee múltiples salidas, cada una de las cuales corresponde a un horizonte distinto. Por lo anterior, la estrategia MIMO emplea solamente un modelo F para las series de tiempo $[y_t, \dots, y_N]$ presentes en $[y_H, \dots, y_{t+1}] = F(y_t, \dots, y_{t-d+1}) + r$ donde $t \in \{d, \dots, N - H\}$, $F : \mathbb{R}^d \rightarrow \mathbb{R}^H$ es un modelo vectorial y $r \in \mathbb{R}^H$ es un vector de ruido, como define Taieb et al. [4]. El pronóstico obtenido por un modelo utilizando la Estrategia MIMO se expresa como:

$$[\hat{y}_{t+H}, \dots, \hat{y}_{t+1}] = \hat{F}(y_N, \dots, y_{N-d+1}), \quad (5)$$

donde N es el paso de tiempo del dato y d es el tamaño de ventana utilizado.

La figura 7 se observa la representación estructural para el modelo MIMO, donde se muestran las entradas y salidas como se mostraron en (5).

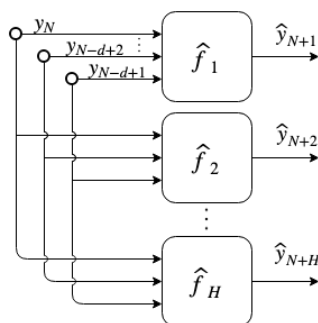


Fig. 6. Estrategia Directa para pronósticos de series de tiempo

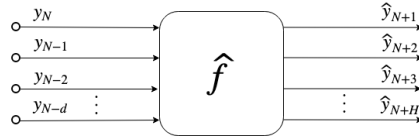


Fig. 7. Estructura de Estrategia MIMO para pronósticos de series de tiempo para H horizonte

4.2. Selección de modelo

Durante la selección de modelo se utilizaron los modelos de *Artificial Neural Networks*: MLP, CNN y LSTM como modelos base, buscando combinar las fortalezas de cada uno de ellos. Las combinaciones realizadas para este trabajo son: MLP, LSTM, LSTM-MLP, CNN-MLP, CNN-LSTM y CNN-LSTM-MLP.

4.3. Preprocesamiento de datos

El objetivo del preprocesamiento es acondicionar los datos para su uso. Dos procesos se usan en esta etapa: **limpieza** y **transformación** de datos.

Limpieza de Datos. El proceso de Limpieza de Datos consiste en explorar la serie de tiempo para buscar datos atípicos, los cuales pueden impactar de forma negativa el proceso y obtener resultados con un error mayor [14]. Este proceso tiene como finalidad reducir el ruido en los datos, e identificar y remover datos atípicos, para este trabajo se utilizan las técnicas de Normalización de Datos y Media Móvil.

Normalización de Datos. El objetivo de la Normalización de Datos es garantizar que los rangos de los valores existentes dentro de la serie de tiempo no sea muy amplio. Esto ayuda a que el proceso de entrenamiento sea más rápido, además de ayudar a reducir el posible sesgo [13]. Para obtener una normalización en el rango $[0, 1]$ se puede usar la expresión

$$N_i = \frac{n_i - \min_n}{\max_n - \min_n}, \quad (6)$$

donde N_i es el valor normalizado obtenido, n_i es el valor original de la serie a normalizar, \min_n es el valor mínimo dentro de la serie y \max_n es el valor máximo.

Media Móvil. La Media Móvil es una función para suavizar o filtrar los datos, ya que reemplaza cada observación y_t con una combinación lineal de los otros datos que están cercanos en el tiempo, tal como lo explica Douglas, et al [12]. Una Media Móvil simple de rango N asigna pesos $1/N$ para las N observaciones mas recientes $Y_T, Y_{T-1}, \dots, Y_{T-N+1}$ y asigna cero a todas las demás observaciones. Lo anterior se puede definir por la expresión:

$$M_T = \frac{Y_T, Y_{T-1}, \dots, Y_{T-N+1}}{N} = \frac{1}{N} \sum_{t=T-N+1}^T y_t, \tag{7}$$

donde M_T es la observación nueva obtenida, Y las observaciones cercanas y N el rango de las observaciones.

Transformación de datos. En el proceso de transformación se obtienen los datos en el formato adecuado para el proceso de entrenamiento del modelo pronosticador. En este trabajo se usan dos formatos, secuencias e imágenes.

Transformación de Serie de Tiempo a Secuencias. La serie de tiempo se transforma en múltiples secuencias de entrenamiento. Como ilustra la figura 8, cada secuencia de entrenamiento tiene longitud $w + H$, con w datos de entrada y H datos esperados. La Tabla 1 muestra la representación de secuencias de una serie de tiempo $X = (x_1, x_2, \dots, x_n)$ y su correspondiente serie normalizada $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$.

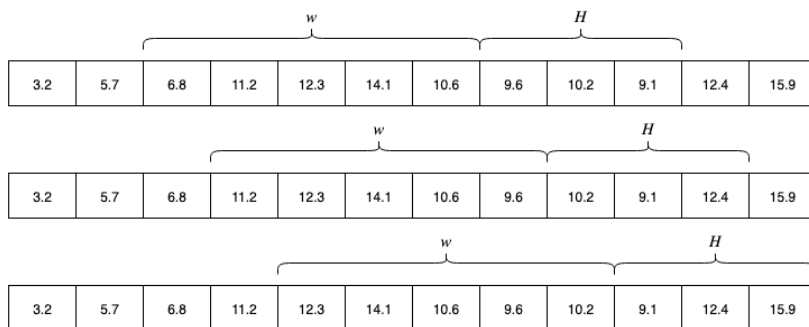


Fig. 8. Ejemplo de generación de secuencias a partir de una serie de tiempo

Tabla 1. Ejemplo de generación de secuencias

# Secuencia	w	Horizonte Esperado
1	$\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \tilde{x}_4, \dots, \tilde{x}_w$	$\tilde{x}_{w+1}, \dots, \tilde{x}_{w+H}$
2	$\tilde{x}_2, \tilde{x}_3, \tilde{x}_4, \tilde{x}_5, \dots, \tilde{x}_{w+1}$	$\tilde{x}_{w+2}, \dots, \tilde{x}_{w+H+1}$
3	$\tilde{x}_3, \tilde{x}_4, \tilde{x}_5, \tilde{x}_6, \dots, \tilde{x}_{w+2}$	$\tilde{x}_{w+3}, \dots, \tilde{x}_{w+H+2}$
4	$\tilde{x}_4, \tilde{x}_5, \tilde{x}_6, \tilde{x}_7, \dots, \tilde{x}_{w+3}$	$\tilde{x}_{w+4}, \dots, \tilde{x}_{w+H+3}$
5	$\tilde{x}_5, \tilde{x}_6, \tilde{x}_7, \tilde{x}_8, \dots, \tilde{x}_{w+4}$	$\tilde{x}_{w+5}, \dots, \tilde{x}_{w+H+4}$
\vdots	\vdots	\vdots
n	$\tilde{x}_n, \tilde{x}_{n+1}, \tilde{x}_{n+2}, \tilde{x}_{n+3}, \dots, \tilde{x}_{w+n}$	$\tilde{x}_{w+n+1}, \dots, \tilde{x}_{w+n+H+1}$

Transformación de de Serie de Tiempo a Imagen. Para obtener una imagen a partir de la serie de tiempo, se utilizó la técnica de *Gramian Angular Field* (GAF) [15]. Cada entrada de la serie se convierte en coordenadas polares con

$$\phi_i = \arccos(\tilde{x}_i), \tag{8}$$

$$r_i = \frac{t_i}{N}. \tag{9}$$

Para identificar la correlación temporal dentro de diferentes intervalos de tiempo se considera la suma trigonométrica entre cada punto [15]. Esto se codifica en la matriz de Gram:

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \dots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \dots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \dots & \cos(\phi_n + \phi_n) \end{bmatrix}. \tag{10}$$

Entre otras propiedades, la matriz de Gram conserva la dependencia temporal de la serie de tiempo. Finalmente, la matriz se convierte en imagen en escala de gris con un escalamiento entre [0, 255]. La figura 9 muestra un ejemplo de una serie de tiempo de 64 datos que se convierte a imagen.

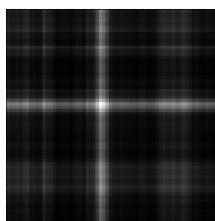


Fig. 9. Ejemplo de Serie de Tiempo a Imagen tamaño de ventana 64

Así como la transformación de series de tiempo a secuencias genera múltiples secuencias, la transformación de series de tiempo a imágenes genera múltiples imágenes. Cada imagen tiene asignado su horizonte (h) esperado. En la Tabla 2 se muestra un ejemplo de lógica que se obtiene al generar un banco de imágenes.

5. Resultados

Los datos utilizados para este trabajo fueron obtenidos por estaciones meteorológicas en distintos puntos de Michoacán, México. Para este trabajo se utilizaron datos de dos localidades diferentes: Aristeo Mercado, y Corrales. Estas series de tiempo contaban con un total de 20,000 datos en intervalos de una hora, de los cuales se utilizó el 30 % para validar el modelo.

Tabla 2. Tabla de ejemplo de generación de Imágenes a partir de secuencias de serie de tiempo

# Imagen	w					<i>Horizonte Esperado</i>		
1	\tilde{x}_1	\tilde{x}_2	\tilde{x}_3	\tilde{x}_4	\dots, \tilde{x}_w	\tilde{x}_{w+1}	\dots	\tilde{x}_{w+H}
2	\tilde{x}_2	\tilde{x}_3	\tilde{x}_4	\tilde{x}_5	\dots, \tilde{x}_{w+1}	\tilde{x}_{w+2}	\dots	\tilde{x}_{w+H+1}
3	\tilde{x}_3	\tilde{x}_4	\tilde{x}_5	\tilde{x}_6	\dots, \tilde{x}_{w+2}	\tilde{x}_{w+3}	\dots	\tilde{x}_{w+H+2}
4	\tilde{x}_4	\tilde{x}_5	\tilde{x}_6	\tilde{x}_7	\dots, \tilde{x}_{w+3}	\tilde{x}_{w+4}	\dots	\tilde{x}_{w+H+3}
5	\tilde{x}_5	\tilde{x}_6	\tilde{x}_7	\tilde{x}_8	\dots, \tilde{x}_{w+4}	\tilde{x}_{w+5}	\dots	\tilde{x}_{w+H+4}
\vdots			\vdots				\vdots	
n	\tilde{x}_n	\tilde{x}_{n+1}	\tilde{x}_{n+2}	\tilde{x}_{n+3}	\dots, \tilde{x}_{w+n}	\tilde{x}_{w+n+1}	\dots	$\tilde{x}_{w+n+H+1}$

Las métricas de error utilizadas son *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE) y *Root Mean Squared Error* (RMSE):

$$\begin{aligned}
 MSE &= \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \\
 MAE &= \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \\
 RMSE &= \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}.
 \end{aligned}
 \tag{11}$$

La Tabla 3 muestra los resultados obtenidos con la serie de tiempo “Aristeo Mercado”. El modelo de IA que obtuvo mejores resultados fue LSTM y la mejor estrategia fue MIMO. En este caso, las LSTM muestran mejor desempeño; sin embargo, se debe considerar que pueden tomar mas tiempo para entrenar. Este factor se debe considerar para estrategias como la Directa donde afecta más el costo computacional.

Tabla 3. Resultados obtenidos con serie de tiempo Aristeo Mercado

Modelo	Estrategia Iterativa				Estrategia Directa				Estrategia MIMO			
	MSE	MAE	RMSE	Tiempo	MSE	MAE	RMSE	Tiempo	MSE	MAE	RMSE	Tiempo
MLP	0.004957	0.053168	0.07040	30.8s	0.004121	0.04845	0.06419	181.14s	0.003716	0.0446845	0.060962	43.9s
LSTM	0.004351	0.050203	0.065965	65.4s	0.003820	0.04543	0.06180	922.53s	0.003702	0.043297	0.060844	84.4s
LSTM-MLP	0.004664	0.051165	0.068295	48.05s	0.003983	0.04652	0.06311	1326.5s	0.003808	0.045964	0.061716	102.1s
CNN-MLP	0.005957	0.059158	0.077182	6.48s	0.004705	0.05031	0.06859	1187.3s	0.004705	0.050317	0.0686	87.9s
CNN-LSTM	0.006317	0.056700	0.079480	78.1s	0.004341	0.04781	0.06588	1438s	0.004443	0.048514	0.06666	134.3s
CNN-LSTM-MLP	0.005795	0.056109	0.076128	60.2s	0.005247	0.05430	0.07243	1328s	0.003989	0.043932	0.063151	219.8s

La Tabla 4 muestra los resultados obtenidos para la serie de tiempo “Corrales”. LSTM con la estrategia MIMO obtienen mejor resultado, seguidos por MLP con estrategia Directa.

En la figura 10 se muestran las gráficas para pronóstico de 24 horas utilizando el mejor modelo de cada serie respectivamente.

Tabla 4. Resultados obtenidos con serie de tiempo Corrales

Modelo	Estrategia Iterativa				Estrategia Directa				Estrategia MIMO			
	MSE	MAE	RMSE	Tiempo	MSE	MAE	RMSE	Tiempo	MSE	MAE	RMSE	Tiempo
MLP	0.004264	0.050060	0.065298	43.62s	0.003570	0.044231	0.059746	124.46s	0.003584	0.043696	0.059865	21.1s
LSTM	0.003963	0.047519	0.062952	36.76s	0.003574	0.044190	0.059782	727.48s	0.0032523	0.04004	0.05673	47.5s
LSTM-MLP	0.004433	0.050148	0.066583	58.34s	0.004392	0.049743	0.066275	1136.07s	0.00326	0.039836	0.05709	75.42s
CNN-MLP	0.005228	0.054755	0.072303	82.57s	0.006229	0.068313	0.078924	1282.26s	0.0038311	0.045463	0.061895	189.89s
CNN-LSTM	0.005272	0.055817	0.072612	156.77s	0.005935	0.068320	0.077042	946.42s	0.0038147	0.046528	0.061763	242.27s
CNN-LSTM-MLP	0.005144	0.054997	0.071724	64.99s	0.006137	0.069617	0.078342	918.9s	0.00542	0.057264	0.07362	137.91s

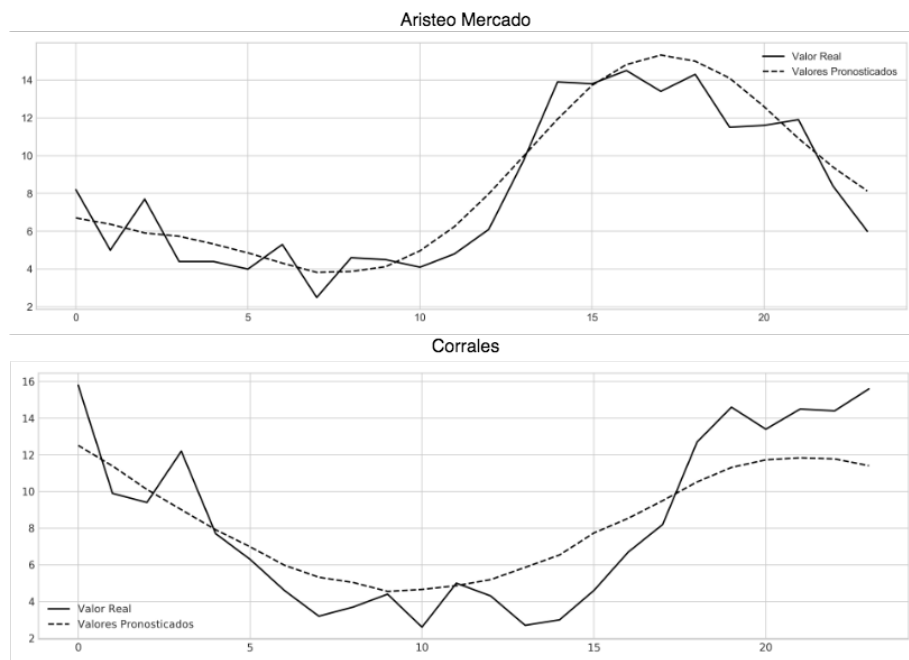


Fig. 10. Gráficas de pronóstico de 24 horas para los mejores resultados en “Aristeo Mercado” y “Corrales”)

La mayoría de los resultados con aptitud deseable fueron aquellos que utilizaron estrategia MIMO y modelo LSTM. Sin embargo, la estrategia Directa y los modelos combinados son más complejos y requieren ajustar más parámetros. Al tener más parámetros se pueden ajustar a diferentes situaciones; sin embargo, del ajuste de parámetros de forma manual requiere mucho tiempo.

6. Conclusiones

El pronóstico de la velocidad de viento a múltiples pasos es una tarea compleja. Sin embargo, utilizar las estrategias y modelos correctos puede facilitar esta tarea. En este trabajo se exploraron tres modelos de Inteligencia Artificial,

además de sus combinaciones y tres estrategias para pronósticos de múltiples pasos. Se observó que el modelo y estrategia con mejor aptitud fueron el modelo LSTM y MIMO, respectivamente. No obstante, usar modelos y estrategias más complejas —tales como la estrategia Directa y los modelos combinados— presenta más dificultad para afinar los modelos porque tienen una gran cantidad de parámetros. Como trabajo futuro se propone combinar el uso de técnicas para optimizar estos parámetros complementando las técnicas presentadas en este trabajo.

Referencias

1. Beltrán Rodríguez, L., Rodríguez Aguilar, J.S. (et. al.): Reporte de Avance de Energías Limpias. Secretaría de Energía (2017)
2. Soman, S.S., Zareipour, H., Malik, O., Mandal, P.: A review of wind power and wind speed forecasting methods with different time horizons. In: North American Power Symposium 2010, pp. 1–8 (2010)
3. An, N.H., Anh, D.T.: Comparison of strategies for multi-step-ahead prediction of time series using neural network. In: 2015 International Conference on Advanced Computing and Applications (ACOMP), pp. 142–149 (2015)
4. Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert systems with applications* 39(8), pp. 7067–7083 (2012)
5. Ahmed, N.K., Atiya, A.F., Gayar, N.E., El-Shishiny, H.: An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, vol. 29, pp. 594–621 (2010)
6. Pasero, E., Raimondo, G., Ruffa, S.: MLP: a multi-layer perceptron application to long-term, out-of-sample time series prediction. In: International Symposium on Neural Networks, pp. 566–575. Springer, Berlin, Heidelberg (2010)
7. Xiaoyun, Q., Xiaoning, K., Chao, Z., Shuai, J., Xiuda, M.: Short-term prediction of wind power based on deep long short-term memory. In: 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), pp. 1148–1152 (2016)
8. Zhu, A., Li, X., Mo, Z., Wu, R.: Wind power prediction based on a convolutional neural network. In: 2017 International Conference on Circuits, Devices and Systems (ICCDs), pp. 131–135, IEEE (2017)
9. Ahmed, A., Khalid, M.: An intelligent framework for short-term multi-step wind speed forecasting based on Functional Networks. *Applied energy*, vol. 225, pp. 902–911 (2018)
10. Qin, Q., Lai, X., Zou, J.: Direct Multistep Wind Speed Forecasting Using LSTM Neural Network Combining EEMD and Fuzzy Entropy. *Applied Sciences*, vol. 9, pp. 126 (2019)
11. Li, Y., Shi, H., Han, F., Duan, Z., Liu, H.: Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy. *Renewable Energy*, vol. 135, pp. 540–553 (2019)
12. Montgomery, D.C., Jennings, C.L., Kulahci, M.: Introduction to time series analysis and forecasting. 2nd Ed. John Wiley & Sons (2015)
13. Nayak, S.C., Misra, B.B., Behera, H.S.: Impact of data normalization on stock index forecasting. *Int. J. Comp. Inf. Syst. Ind. Manag. Appl*, vol. 6, pp. 357–369 (2014)

14. Tsay, R.S.: Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, vol. 7, pp. 1–20 (1988)
15. Wang, Z., Oates, T.: Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In: *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015)