

Pronóstico de series de tiempo de imágenes de sequías utilizando autocodificadores y redes neuronales

Manuel Medrano, Juan Flores, Héctor Rodríguez,
Rodrigo López, Carlos Lara

Tecnológico Nacional de México,
Instituto Tecnológico de Culiacán,
México

manuel.medrano@itculiacan.edu.mx

Resumen. Las sequías son uno de los peligros climáticos que más afectan a las poblaciones. Estas son causadas en su mayoría debido por el calentamiento global. De esta manera, grandes zonas de población pueden quedarse sin agua durante un periodo de tiempo indefinido. Un pronóstico correcto de este tipo de eventos puede permitir a grandes poblaciones a prepararse para contrarrestar estas situaciones. Tradicionalmente el pronóstico de series de tiempo se realiza mediante un conjunto de datos numéricos con el cual se puedan encontrar patrones de comportamiento de la información dentro de los siguientes n espacios de tiempo. Este trabajo propone una nueva metodología para predecir series de tiempo de imágenes basada en autoencoders (autocodificadores) aplicada al estudio de las sequías. Utilizando una serie de tiempo en formato de imagen, es posible la obtención de los siguientes pasos de tiempo, y retornar la información de pronóstico como una imagen que representa el futuro, mediante la combinación de modelos de tipo autoencoder y múltiples redes neuronales.

Palabras clave: Series de tiempo, autoencoder, redes neuronales, generador de imágenes.

Drought Imaging Time Series Forecasting Using Autoencoders and Neural Networks

Abstract. Droughts are one of the climatic hazards that most affect populations. These are mostly caused by global warming. In this way, large population areas can be left without water for an indefinite period of time. A correct forecast of this type of event can allow large populations to prepare to counteract these situations. Traditionally, time series forecasting is done using a set of numerical data with which patterns of behavior of the information can be found within the following n periods of time. This work proposes a new methodology to predict image time series based on autoencoders (autocoders) applied to the study of droughts. Using a time series in image format, it is possible to obtain the following time steps, and return the forecast information as an image that represents the future, by combining autoencoder-type models and multiple neural networks.

Keywords: Time series, autoencoder, neural networks, image generator.

1. Introducción

En años recientes las sequías se han visto aumentadas en el mundo, debido al cambio climático. Este evento climático afecta a más cantidad de personas que cualquier otro peligro climático [9]. En el trabajo de Kallis [9] describen a las sequías como una falta temporal de agua, que es necesariamente pero no explícitamente causada por una anomalía climática y que daña a una actividad, a grupos o al medio ambiente.

Algunos de los problemas causados por las sequías son: Reducción de cultivos, pastizales, productividad forestal, niveles de agua superficiales y subterráneas, así como aumento de incendios y tasa de mortalidad de ganado y vida silvestre. De igual manera este evento climático tiene repercusiones en cuanto al suministro de agua y alimentos en la actividad económica.

En este trabajo se aborda la estimación de las sequías mediante el pronóstico de series de tiempo de imágenes. El pronóstico de series de tiempo es una técnica estadística utilizada comúnmente para la estimación de valores en el futuro a partir de datos históricos [16]. Mediante la utilización de información estructurada, que representa información histórica tomada en periodos regulares de tiempo, y la utilización de técnicas de pronóstico, se puede obtener un conjunto de valores que representan la información futura.

Gracias al pronóstico de series de tiempo se pueden obtener diferentes beneficios. Uno de los usos que generalmente se le otorga a esta técnica es la toma de decisiones. Debido a que el pronóstico de series de tiempo muestra valores que aun no han sido presentados, se puede tomar una acción inmediata, en diferentes áreas y con diferente nivel de impacto, como: capacidad de producción, demanda de producción, pronóstico de clima, sequías, inundaciones, entre otros.

De esta manera se pueden tomar decisiones tanto para mejorar como para combatir amenazas. Por ejemplo, en el trabajo [17] muestran, mediante técnicas de pronóstico, una forma de prevenir inundaciones repentinas observando las descargas máximas en el pasado. Comúnmente el tipo de información de series de tiempo que es utilizada para el pronóstico es representada con un vector.

Otra representación alternativa es el formato de imagen o en forma de matriz. Siguiendo el mismo procedimiento que las series de tiempo, las imágenes son tomadas periódicamente en un intervalo constante de tiempo. De tal manera que, al igual que con la información numérica, sea posible hacer un pronóstico de una serie de imágenes, y dicho pronóstico representarán una estimación futura.

Los enfoques comunes pueden ser los siguientes: 1) Obtener información relevante de un conjunto de imágenes para representarlas en series de tiempo con formato numérico, y trabajar con esta información, como lo muestra en [18]; 2) Transformar un conjunto de series de tiempo con formato numérico a un conjunto de imágenes y que estas sean evaluadas por un proceso que pueda manejarlas en este formato, tal como se realiza en [10].

Este trabajo propone la compresión de la serie de tiempo de imágenes mediante el empleo de un autoencoder para la reducción del espacio de búsqueda. Con la información codificada generamos n redes neuronales (una red por cada píxel contenida en la imagen codificada).

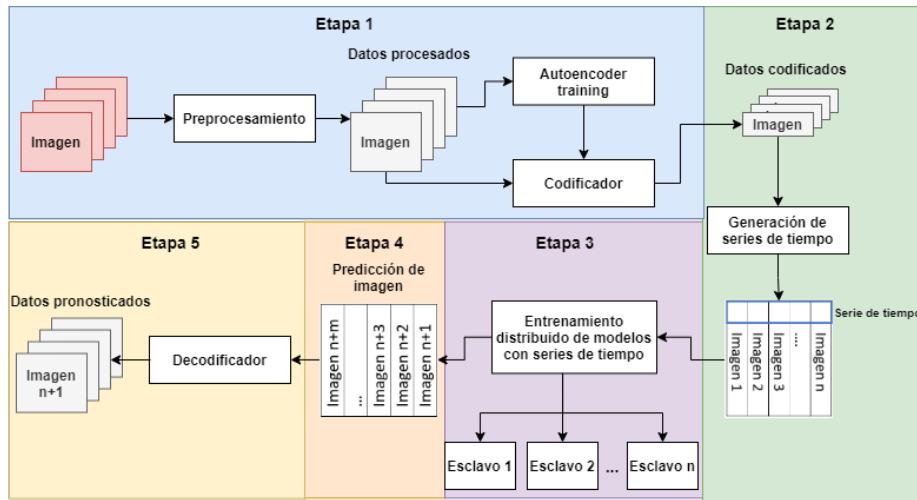


Fig. 1. Procedimiento de pronóstico de imágenes propuesto.

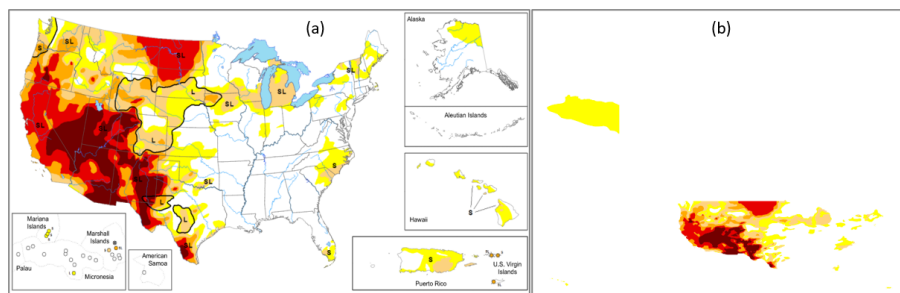


Fig.2. a) Mapa original de sequías. b) Representación de la sequía de EU. Obtenidos del sitio [15].

Para posteriormente generar el siguiente instante en el tiempo (imagen codificada). Finalmente, decodificamos la imagen generada por las n redes neuronales para obtener la representación real. En este trabajo planteamos otra aproximación para solucionar el pronóstico de series de tiempo utilizando un conjunto de imágenes tomadas periódicamente que representan zonas de sequías, esta información es de tipo “Geographic Information System”(GIS) en formato “Web Map Service” (WMS) [15].

Considerando que este conjunto de imágenes puede ser tratado como una serie de tiempo, es válido afirmar que, utilizando directamente este conjunto de imágenes, se pueda generar la información histórica futura en el mismo formato de imagen. De tal manera que, la imagen de salida del proceso sea una imagen totalmente nueva muy similar a la imagen original y, que es pronosticada por un conjunto de modelos de aprendizaje máquina.

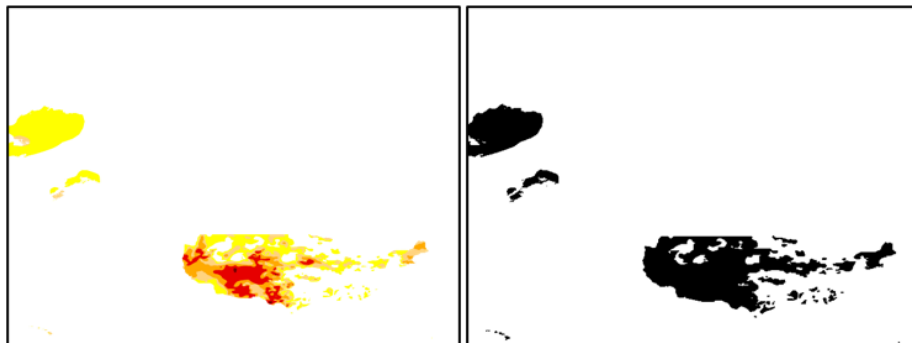


Fig. 3. Procesamiento de imágenes de color a imagen binaria.

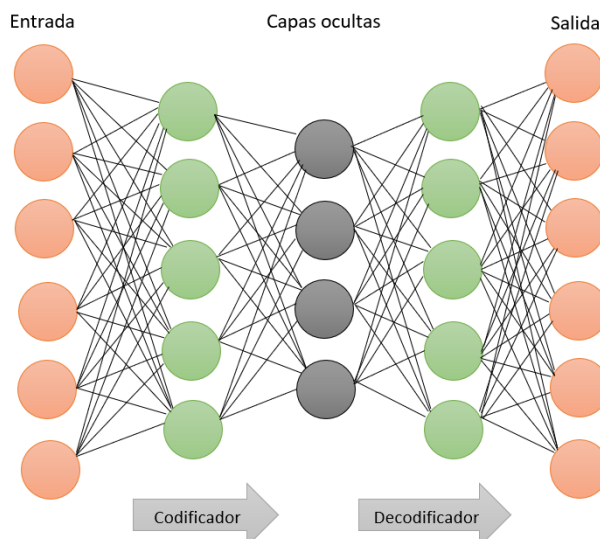


Fig. 4. Arquitectura de una red neuronal de autoencoder. Tomado de [8].

2. Trabajo relacionado

El pronóstico de series de tiempo se puede encontrar en diferentes áreas, como puede ser la Economía, Energía, Medicina o la Ingeniería [7]. Mediante la utilización de diversas técnicas, se pueden obtener estimaciones minimizando la incertidumbre lo más posible [1]. En el trabajo [5] se presentan, discuten y evalúan una serie de técnicas para pronósticos de series de tiempo, estos modelos son:

Redes neuronales artificiales (ANN), Media Móvil Integrada Auto-Regresiva (ARIMA), Maquinas de Soporte Vectorial (SVM), Razonamiento basado en casos (CBR), Series de tiempo difusas, Modelo de predicción gris, Media Móvil y Suavisado Exponencial (MA & ES), K-Vecinos cercanos (KNN) y Modelos híbridos.

Tabla 1. Niveles de codificación, muestra original 480×640 .

| Conjunto de capas | Dimensiones de codificación | Total de datos |
|-------------------|-----------------------------|----------------|
| 2 | (120, 160, 8) | 153,600 |
| 3 | (60, 80, 4) | 19,200 |
| 4 | (30, 40, 4) | 4,800 |
| 5 | (15, 20, 4) | 1,200 |

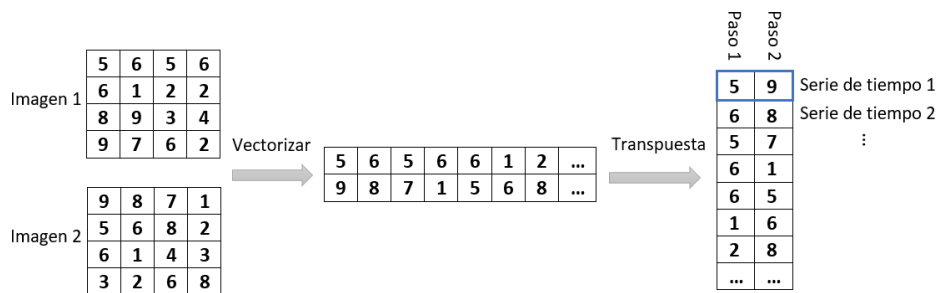


Fig. 5. Generación de series de tiempo por imágenes codificadas.

Concluyendo que la utilización de las técnicas antes mencionadas ofrecen una serie de ventajas y desventajas, y que hacen un énfasis en la utilización de los modelos híbridos, pues estos permiten tomar ventajas de las fortalezas y debilidades de cada uno de los modelos. Existen diferentes enfoques en cuanto al pronóstico de series de tiempo. Muchos de estos enfoques utilizan las características distintivas de las imágenes, definidas por la aplicación dada, para realizar un pronóstico basado en la secuencia de imágenes.

Por ejemplo, un enfoque básico de pronóstico es utilizar dos imágenes consecutivas para pronosticar un vector de características. Como se realiza en [12] donde se asume que todos los objetos de las imágenes se trasladan en la misma dirección, y de esta manera encuentra el mejor vector que representa dicha traslación. Otra suposición de este enfoque básico puede ser que las imágenes no son consecutivas, pero los píxeles se intensifican y permanecen constantes al pasar del tiempo.

En [3] utilizan un enfoque basado en la estimación de vectores de movimiento por el método de correlación cruzada, mediante la partición de las imágenes en subconjuntos de píxeles del mismo tamaño y asigna el vector que une los subconjuntos de imágenes consecutivas con el mayor coeficiente de correlación cruzada. En cambio, en [14] se maximiza la probabilidad del campo vectorial dadas dos imágenes consecutivas, mediante la utilización de la metaheurística del recocido simulado.

Otros enfoques utilizan los modelos de regresión. Como en [19] donde se realiza un análisis de sensibilidad de la respuesta de los parámetros de crecimiento de la altura del tallo, el índice de área foliar y la biomasa al número de días después de la siembra. También es posible utilizar Máquinas de Soporte Vectorial (SVM), enfocándose en determinar si se pueden separar los puntos de datos con un hiperplano $(k - 1)$ dimensional.

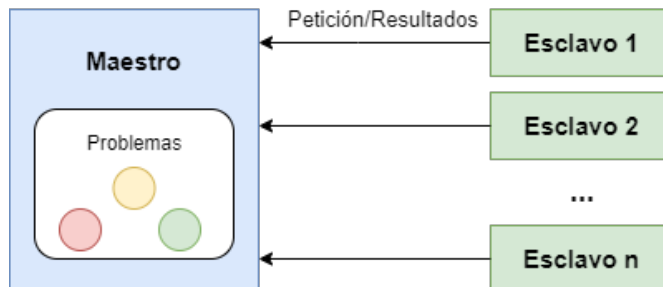


Fig. 6. Metodología maestro-esclavo.

Tabla 2. Niveles de codificación, muestra original 480×640 .

| Forma | Redes neuronales | Tiempo | Error autoencoder | Precisión paso 1 | Error paso 1 |
|-------------|------------------|----------------|-------------------|------------------|--------------|
| (15,20,4) | 1,200 | 0d-0h-52m-58s | 0.0343 | 0.9561 | 0.0439 |
| (30,40,4) | 4,800 | 0d-3h-28m-51s | 0.0187 | 0.9682 | 0.0318 |
| (60,80,4) | 19,200 | 0d-14h-10m-41s | 0.0163 | 0.9805 | 0.0195 |
| (120,160,8) | 153,600 | 4d-17h-40m-18s | 0.0059 | 0.9849 | 0.0151 |

En [4] utilizan este enfoque mediante el análisis de la evolución espacio-temporal de la superficie del mar, extrayendo las series de tiempo de imágenes de radar. Algunos enfoques utilizan técnicas de aprendizaje automático, como puede ser las Redes Neuronales Artificiales (ANNs). Las ANNs son una aproximación que puede ser utilizada en una amplia gama de problemas relacionados con las series de tiempo.

Por ejemplo, en [13] se utiliza una ANN para el pronóstico de la irradiancia global horizontal dadas dos entradas: campos vectoriales de movimiento de nubes y imágenes de índice de nubes. Otro enfoque que ha sido utilizado de manera exitosa es el aprendizaje profundo. En este campo los modelos comúnmente utilizados son las Redes Neuronales de Convolución (CNN) para el manejo de imágenes y las Redes de memoria a corto plazo (LSTM) para el manejo de secuencias correlacionadas.

En [11] se transforma una red de usuarios de metro en imágenes, obteniendo así con una CNN la información espacial de los datos. De manera separada, los datos de la serie temporal se introducen en un modelo LSTM que extrae de manera apropiada las características temporales, para luego combinar la información espacial y las características espaciales con una red ANN que realiza el proceso de pronóstico del número de pasajeros del metro. Con lo anterior se puede concluir que la utilización de arquitecturas híbridas presentan buenos resultados si se utilizan de manera adecuada en una parte específica de la arquitectura.

De esta manera, en este trabajo se realiza una implementación híbrida con múltiples arquitecturas en diferentes puntos. Utilizando un autoencoder, para la codificación de imágenes, se busca la reducción de la cantidad de características necesarias para su implementación. En conjunto con la utilización de una arquitectura que utiliza LSTMs, para la extracción de características temporales, junto con ANNs, para la extracción de características y pronósticos.

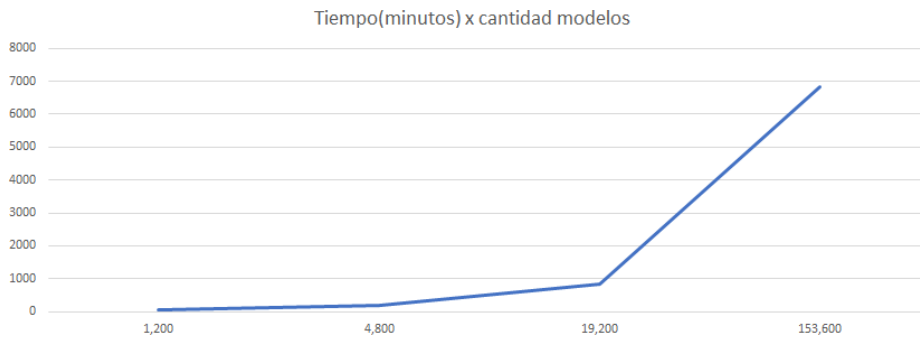


Fig. 7. Relación cantidad de modelos por tiempo de ejecución.

3. Materiales y métodos

Para realizar la estimación del siguiente instante de tiempo (imagen) dada una secuencia de imágenes consecutivas se realizan en 5 pasos o etapas. La primera etapa consiste en comprimir (codificar) la información mediante el empleo de autoencoders. La segunda etapa utiliza la información codificada (comprimida), la vectoriza y genera las n series de tiempo (donde n es igual al número de píxeles en cada imagen codificada).

La tercera etapa genera y entrena n redes neuronales (una por cada píxel contenido en la imagen codificada). La cuarta etapa utiliza las redes entrenadas y genera la siguiente imagen (codificada) en el tiempo. Para finalmente la quinta etapa utilizando en autoencoder previamente definido de codifica la imagen generada por las n redes neuronales. La Figura 1 muestra el flujo del proceso principal del pronóstico de imágenes.

3.1. Datos

Los datos utilizados en este trabajo fueron tomados de una página web que mantiene un monitor de las sequías en Estados Unidos semana con semana. La información fue tomada por un grupo de expertos que a partir de esta información la transforma en un mapa que muestra las regiones de Estados Unidos que se encuentran en una sequía.

El mapa utiliza cinco categorías, cada una de estas categorías muestra su grado de impacto en las diferentes áreas, estas categorías son: D0 anormalmente seco, esto es una sequedad a corto plazo que detiene o hace más lento el crecimiento de cultivos/pastizales; D1 sequía moderada, algunos cultivos/pastizales han sufrido daños, y las fuentes, corrientes, embalses o posos presentan signos de escasez; D2 Sequía severa, posibilidad de pérdidas en cultivos/pastizales y la escasez de agua es común.

D3 Sequía extrema, hay pérdidas significantes de cultivos/pastizales y hay restricciones de agua extendidos; D4 Sequía excepcional, hay pérdidas excepcionales de cultivos/pastizales y escasez de agua en fuentes, corrientes, embalses o posos [15]. En este trabajo se utiliza información desde el 4 de Enero del 2000 hasta el 1 de Agosto del 2020. Obteniendo así un conjunto de datos de 1,079 muestras para utilizar.

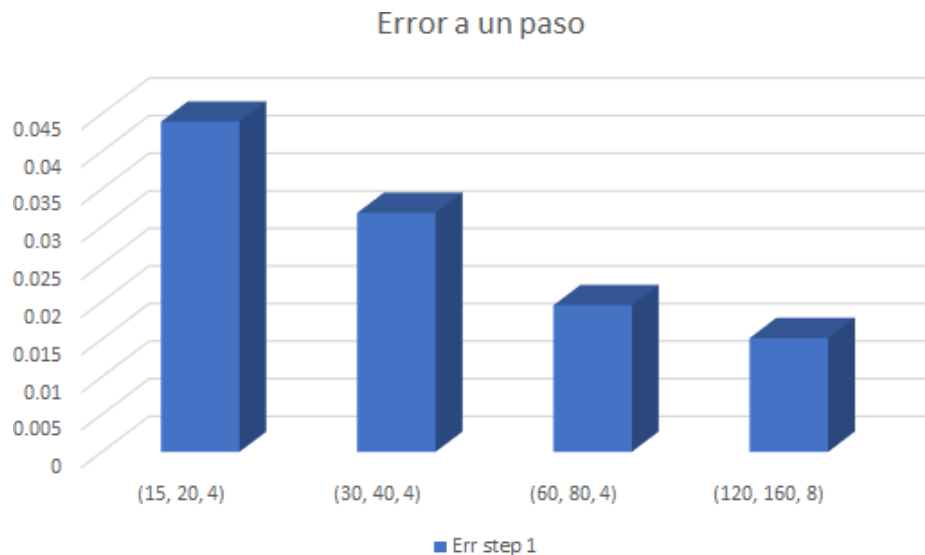


Fig. 8. Error en diferentes niveles de codificación.

La información utilizada es de tipo "Geographic Information System"(GIS) mediante un servicio "Web Map Service"(WMS). Este servicio provee imágenes de mapas georeferenciados que contiene la información limpia de delimitaciones. Cada archivo WMS contiene una capa del Monitor de Sequía basada en los shapefiles limpios y re-proyectados a la proyección WGS84 [15]. La Figura 2 muestra un ejemplo de la representación de la sequía en EU.

3.2. Preprocesamiento de las imágenes

Cuando se habla de pronósticos, se habla de reconocimiento de patrones. Estas técnicas engloban una serie de problemas que las hace tener algún grado de dificultad para su implementación. Muchas de las dificultades básicas del aprendizaje máquina se derivan principalmente del uso de grandes cantidades de entradas. De manera que, las tareas se vuelven exponencialmente más difíciles a medida de que el número de entradas aumenta.

A este problema se le conoce como la "Maldición de la dimensionalidad" [8]. En este caso, las imágenes originales son de 480×640 en RGB lo cual da un total de 921,600 valores por cada una de las imágenes del conjunto de datos. Esto representa un problema, pues el manejo de esta gran cantidad de valores requerirá tanto mayor poder de procesamiento como tiempo de ejecución. Para evitar este problema, se utilizan imágenes con un conjunto de colores diferente.

Al utilizar las imágenes con una gama de colores tipo escala de grises maneja un solo canal de color, lo cual reduce la cantidad de valores por imágenes de 307,200 valores. Sin embargo, la cantidad de colores que tiene que aprender a diferenciar los modelos de aprendizaje máquina sigue siendo grande.

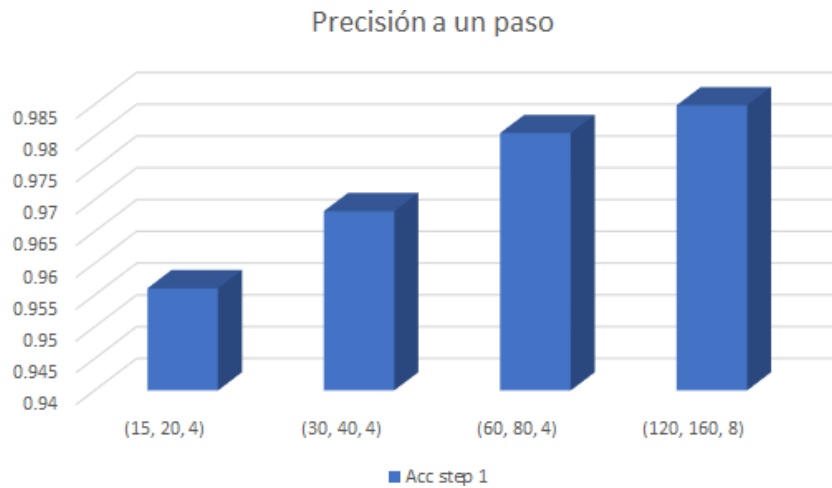


Fig. 9. Precisión en diferentes niveles de codificación.

Como la escala de grises maneja un total de 255 valores diferentes, el modelo tiene que ser capaz de identificar correctamente toda esta gama de colores. Para reducir esto, se utiliza una escala binaria de colores, que transforma las imágenes a solo 2 valores diferentes, blanco o negro. De esta manera, las imágenes respetan otro tipo de información, hay sequía o no hay sequía. La Figura 3 muestra una comparación con la imagen original y la imagen transformada en color monocromático.

3.3. Codificación de las imágenes

En [8] describen a un autoencoder como un tipo de red neuronal que trabaja sin etiquetas y que pertenece a la clase del aprendizaje no supervisado. Este tipo de red contiene una capa de entrada que coincide con las dimensiones de la imagen, o vector de entrada, junto con capas ocultas las cuales reducen la dimensionalidad, seguido de una capa de salida con la mismas dimensiones que la entrada.

El objetivo de esta arquitectura es codificar y decodificar la información introducida en la capa de entrada, la imagen original, en la capa de salida, donde en el proceso de codificación se reduce la dimensionalidad de la imagen, y en la decodificación esta se reconstruye. La Figura 4 muestra la arquitectura de un autoencoder. Tal como se muestra en la Figura 4 un autoencoder esta constituido por dos partes: el codificador, que se encarga de reducir la dimensionalidad de la información; y el decodificador, que se encarga de aumentar la dimensionalidad hasta el punto original.

En este trabajo se utilizan estas dos partes del autoencoder para reducir la dimensionalidad de la información y que sea mucho más fácil realizar un pronóstico de las sequías. Los autoencoders son un conjunto de capas de Convolución y Max pooling, donde las capas de Max pooling se encarga de reducir apropiadamente las primeras dos dimensiones, y las capas de convolución define la tercera dimensión de las muestras. Dependiendo del número de conjuntos de capas que se utilicen es la dimensión resultante en la capa intermedia.



Fig. 10. Imagen pronosticada con forma de codificación (15, 20, 4).

La muestra comprimida por este modelo de red neuronal es en realidad una representación de la muestra original pero con las dimensiones reducidas. De esta manera, en este trabajo se utiliza el conjunto de datos codificados para reducir la necesidad de poder y tiempo computacional. La Tabla 1 muestra el conjunto de capas Convolución-Maxpooling utilizadas, las dimensiones de las imágenes que se obtienen después de la codificación y el total de datos por imagen resultante de esta codificación.

3.4. Generación de series de tiempo

En esta sección se presenta la transformación de las imágenes codificadas a vectores de series de tiempo. Cada píxel de un imagen codificada representa una serie de tiempo independiente. De modo que, al juntar los píxeles de todas las imágenes se obtiene múltiples conjuntos de series de tiempo. Donde cada uno de estos conjuntos de datos pueda ser utilizados como información de entrada para un solo modelo.

De esta manera, cada uno de los conjuntos de datos es tratado como una serie de tiempo independiente, con el cual se realiza un proceso común de pronóstico de series de tiempo. Así cada uno de los modelos que hayan sido entrenados pronosticará una parte de una imagen codificada que represente el siguiente paso en el tiempo. De tal manera que, la imagen codificada pase por el decodificador y se genere una imagen del siguiente paso en la secuencia.

Para realizar este proceso el primer paso consiste en transformar todo el conjunto de datos de múltiples dimensiones a un solo vector de una dimensión. Posteriormente juntar cada uno de estos vectores de manera adyacente, y realizar una transpuesta a la información para una mejor manipulación. De esta manera cada una de las filas del nuevo conjunto de datos obtenido, representara una parte de cada una de las imágenes, osea una secuencia de series de tiempo. La Figura 5 muestra un ejemplo de los pasos a seguir de este proceso.

3.5. Entrenamiento distribuido

Debido principalmente a que cada una de las series de tiempo antes descritas va a ser utilizada por un modelo diferente, serán necesarios miles de modelos de aprendizaje máquina para realizar los pronósticos.



Fig. 11. Imagen pronosticada con forma de codificación (30, 40, 4).

Lo anterior representa una gran cantidad de tiempo invertido si fuera a ser realizado por una sola máquina. En este trabajo se optó por utilizar una técnica de sistemas distribuidos donde todos estos entrenamientos serán realizados en múltiples computadoras al mismo tiempo.

En [2] describen que las aplicaciones de sistemas distribuidos están compuestas por múltiples aplicaciones diferentes que se ejecutan en máquinas diferentes, o muchas replicas ejecutándose a través de diferentes máquinas, todas comunicándose juntas para implementar un sistema.

Utilizando un modelo maestro-esclavo, se distribuyen las responsabilidades entre estos nodos. En el trabajo [6] se define este método se como un modelo simple basado en una política de p esclavos a p conjuntos, donde los individuos son separados en p conjuntos iguales y enviados a p esclavos para su evaluación.

Según el trabajo [6] se asume que la arquitectura maestro-esclavo sigue un paradigma cliente-servidor donde las conexiones se cierran después de cada solicitud. De esta manera, un esclavo (cliente) se conecta a un maestro (servidor) para solicitar un trabajo, el maestro responde enviando un conjunto de tareas que son definidas por el maestro.

La Figura 6 muestra una representación de esta metodología. Durante la ejecución, las tareas del maestro corresponden a la generación, entrenamiento y validación de un autoencoder, codificación del conjunto de datos, creación del conjunto completo de series de tiempo, partición equitativa de cantidad de entrenamiento por esclavo, generación de pronósticos y decodificación del pronóstico.

Las tareas del esclavo corresponden al manejo de lectura de parámetros de entrenamiento, que son: tamaño de ventana de series de tiempo, índice inicial y final de los conjuntos de datos, y la carpeta de almacenamiento de modelos. Generación de cada una de las series de tiempo, creación y entrenamiento de los modelos de pronóstico, modelo de dos capas LSTM y dos ANN y una capa de salida, y guardado del modelo entrenado.

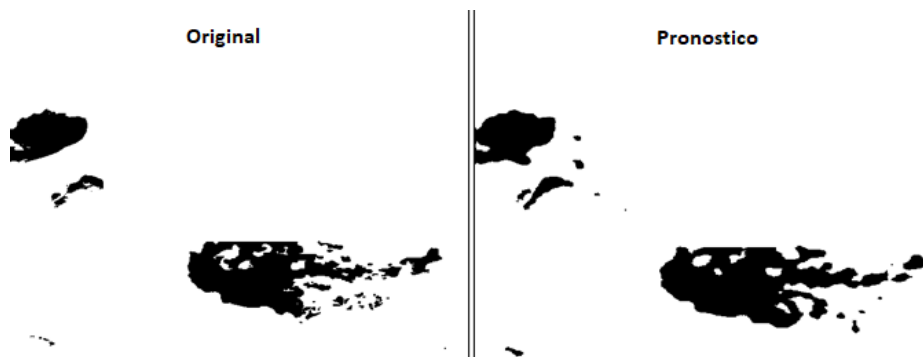


Fig. 12. Imagen pronosticada con forma de codificación (60, 80, 4).

4. Resultados

Para probar la efectividad de la metodología propuesta, fueron realizados diferentes experimentos. Los experimentos fueron realizados con dos esclavos en un solo equipo de computadora, en el lenguaje de programación python, con las siguientes características: CPU Intel Core i7-8700, 16 GB de RAM y dos NVIDIA GeForce RTX 2080 TI. Cada esclavo utiliza exclusivamente una sola GPU.

Ambos esclavos utilizan el mismo modelo de red neuronal, y ese mismo modelo es utilizado para cada una de las series de tiempo, el modelo consiste en: Una capa de entrada de tipo LSTM, una capa oculta de tipo LSTM, dos capas ocultas de tipo ANN y una capa de salida.

Todas las capas utilizan la función de activación “elu”, el modelo utiliza la función de pérdida Error Cuadrático Medio (MSE) y función de optimización rmsprop”. Cada una de las redes son entrenadas con un Batch Size de 64 por 150 épocas con un Early Stopping con un valor de paciencia de 15.

La Tabla 2 muestra una comparativa con las diferentes características de los experimentos realizados. Donde la primera columna muestra la forma de compresión del experimentos, la segunda columna muestra el total de redes neuronales implementadas y la tercera columna el tiempo total que llevo realizar el experimento.

A partir de la cuarta columna se muestran los valores del entrenamiento, estos son el valor de pérdida del autoencoder, y los valores de precisión y error de las imágenes pronosticadas del paso uno y paso dos. Los valores de precisión y pérdida de las imágenes fueron obtenidos de la Ecuación 1 y 2 correspondientemente:

$$\frac{\sum_{i=1}^n \sum_{j=1}^m 1 - |\text{original}_{ij} - \text{pronostico}_{ij}|}{nm}, \quad (1)$$

$$\frac{\sum_{i=1}^n \sum_{j=1}^m |\text{original}_{ij} - \text{pronostico}_{ij}|}{nm}, \quad (2)$$

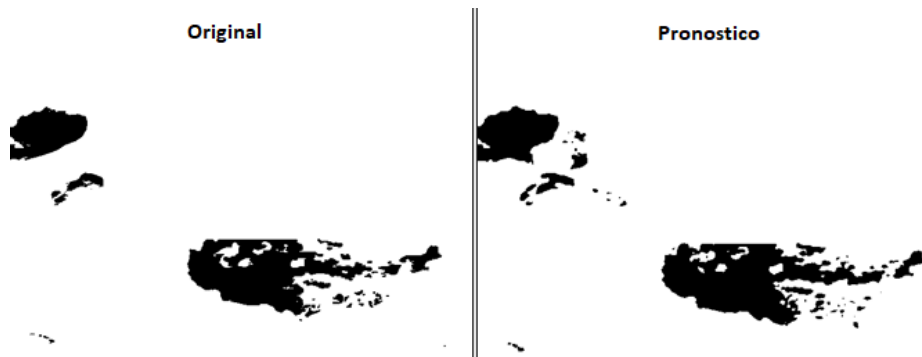


Fig. 13. Imagen pronosticada con forma de codificación (120, 160, 8).

donde n es la anchura de la imagen y m es la altura de la imagen. Con las Ecuaciones 1 y 2 se obtiene el número de coincidencias entre la imagen original y la pronosticada, es decir la comparación de que tantos píxeles son iguales. Para complementar los valores obtenidos en la Tabla 1, la Figura 7 muestra una gráfica del tiempo de entrenamiento. En donde se compara la cantidad de tiempo invertido con la cantidad de modelos a implementar.

En la Tabla 7 se puede observar que al tener un menor nivel de codificación, es decir, una mayor cantidad de modelos a implementar, se requiere una inversión de tiempo mayor a la hora de implementar esta metodología. Por ejemplo, al utilizar un nivel bajo de codificación con el autoencoder, se obtienen dimensiones de codificación muy grandes, como lo muestra la primera fila de la Tabla 1 que tiene las dimensiones (120, 160, 8), dando un total de 153,600 valores.

Como muestra la Figura 7 el tiempo necesario para la implementación de estas dimensiones de codificación es muy alto y no hay una ganancia significativa en cuanto a precisión y error que justifique utilizar niveles tan bajos de codificación, tal como lo muestran la Figura 9 y 8.

Después de pasar por todo el proceso de entrenamiento, cada uno de los modelos puede pronosticar n pasos hacia el futuro, es decir un píxel de una imagen. Para después pasar por el autoencoder entrenado y decodificar esa imagen pronosticada. La Figura 10, 11, 12 y 13 muestran una comparación con los pasos futuros originales y los pronosticados.

5. Conclusiones

Para obtener una imagen que represente los siguientes pasos en el tiempo de las sequías, es necesario un proceso extenso. Este proceso requiere diversos pasos que son necesarios para su implementación. El paso que requiere más tiempo de implementación es el entrenamiento de la gran cantidad de modelos necesarios para llevar a cabo el pronóstico. Pero aunque se requiera mucho tiempo, este puede ser reducido si se implementan una cantidad mayor de trabajadores que puedan realizar los pronósticos de manera simultanea en paralelo.

En los resultados mostrados en la Figura 9 y 8, se puede observar que a partir de la codificación con dimensiones (60,80,4) no se obtiene una ganancia significativa de precisión y error que valga la pena para implementación a niveles de codificación más bajos. Mediante el uso del valor de error del entrenamiento del autoencoder, se podría encontrar un punto mínimo de codificación el cual ayudaría a obtener buenos resultados en un tiempo menor.

Los experimentos realizados fueron un gran reto al momento de producir los pronósticos, principalmente debido a la gran cantidad de recursos necesarios para su implementación. Los experimentos más extensos utilizaban alrededor de los 80 Gb de espacio para almacenar todos los modelos entrenados necesarios. Igualmente el tiempo necesario para producir un pronóstico adecuado de las imágenes puede llegar a tomar varios días de ejecución dependiendo de los recursos disponibles.

Si bien aplicar esta metodología puede ser un poco difícil, debido a la necesidad de recursos y la complejidad de la misma, los resultados obtenidos fueron positivos conforme al pronóstico de las sequías. Después de todo el proceso los pronósticos alcanzaron un 98 % de precisión en algunos casos, obteniendo imágenes muy fieles a las originales.

Referencias

1. Brownlee, J.: Deep learning for time series forecasting: Predict the future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery (2018)
2. Burns, B.: Designing distributed systems: Patterns and paradigms for scalable, reliable services. O'Reilly Media, Inc. (2018)
3. Chow, C.W., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., Washom, B.: Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. Solar Energy, vol. 85, no. 11, pp. 2881–2893 (2011) doi: 10.1016/j.solener.2011.08.025
4. Cornejo-Bueno, L., Nieto Borge, J. C., Alexandre, E., Hessner, K., Salcedo-Sanz, S.: Accurate estimation of significant wave height with support vector regression algorithms and marine radar images. Coastal Engineering, vol. 114, pp. 233–243 (2016) doi: 10.1016/j.coastaleng.2016.04.007
5. Deb, C., Zhang, F., Yang, J., Lee, S. E., Shah, K. W.: A review on time series forecasting techniques for building energy consumption. Renewable and Sustainable Energy Reviews, vol. 74, pp. 902–924 (2017) doi: 10.1016/j.rser.2017.02.085
6. Dubreuil, M., Gagne, C., Parizeau, M.: Analysis of a master-slave architecture for distributed evolutionary computations. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 36, no. 1, pp. 229–235 (2006) doi: 10.1109/tsmcb.2005.856724
7. Gamboa, J. C.: Deep Learning for Time-Series Analysis (2017) doi: 10.48550/ARXIV.1701.01887
8. Joshi, A. V.: Machine learning and artificial intelligence. Springer International Publishing (2020) doi: 10.1007/978-3-030-26622-6
9. Kallis, G.: Droughts: Annual review of environment and resources. vol. 33, no. 1, pp. 85–118 (2008) doi: 10.1146/annurev.enviro.33.081307.123117
10. Li, X., Kang, Y., Li, F.: Forecasting with time series imaging. Expert Systems with Applications, vol. 160, pp. 113680 (2020) doi: 10.1016/j.eswa.2020.113680
11. Ma, X., Zhang, J., Du, B., Ding, C., Sun, L.: Parallel architecture of convolutional bi-directional LSTM neural networks for network-wide metro ridership prediction. In: IEEE

- Transactions on Intelligent Transportation Systems, vol. 20, no. 6, pp. 2278–2288 (2019) doi: 10.1109/tits.2018.2867042
12. Magnone, L., Sossan, F., Scolari, E., Paolone, M.: Cloud motion identification algorithms based on all-sky images to support solar irradiance forecast. In: Proceedings of the IEEE 44th Photovoltaic Specialist Conference (2017) doi: 10.1109/pvsc.2017.8366102
 13. Marquez, R., Pedro, H. T. C., Coimbra, C. F. M.: Hybrid solar forecasting method uses satellite imaging and ground telemetry as inputs to ANNs. *Solar Energy*, vol. 92, pp. 176–188 (2013) doi: 10.1016/j.solener.2013.02.023
 14. Mohamed Sallah, A. H., Tychon, B., Piccard, I., Gobin, A., Van Hoolst, R., Djaby, B., Wellens, J.: Batch-processing of AquaCrop plug-in for rainfed maize using satellite derived fractional vegetation cover data. *Agricultural Water Management*, vol. 217, pp. 346–355 (2019) doi: 10.1016/j.agwat.2019.03.016
 15. United States Drought Monitor: El U.S. Drought Monitor es un producto de la colaboración entre el National Drought Mitigation Center de University of Nebraska-Lincoln, el United States Department of Agriculture y el National Oceanic and Atmospheric Administration (2021) droughtmonitor.unl.edu/Data/GISData.aspx
 16. Wei, W. W.: Time series analysis. Oxford University Press (2013) doi: 10.1093/oxfordhb/9780199934898.013.0022
 17. Yan, W., Liu, J., Zhang, M., Hu, L., Chen, J.: Outburst flood forecasting by monitoring glacier-dammed lake using satellite images of Karakoram Mountains. *Quaternary International*, vol. 453, pp. 24–36 (2017) doi: 10.1016/j.quaint.2017.03.019
 18. Yang, L., Gao, X., Li, Z., Jia, D., Jiang, J.: Nowcasting of surface solar irradiance using FengYun-4 satellite observations over China. *Remote Sensing*, vol. 11, no. 17, pp. 1984 (2019) doi: 10.3390/rs11171984
 19. Zhang, W., Chen, E., Li, Z., Zhao, L., Ji, Y., Zhang, Y., Liu, Z.: Rape (*Brassica napus* L.) growth monitoring and mapping based on radarsat-2 time-series data. *Remote Sensing*, vol. 10, no. 2, pp. 206 (2018) doi: 10.3390/rs10020206