



Series temporales para el índice Diferencial Normalizado de Vegetación mediante una Red Neuronal Artificial de corto y largo plazo, y el algoritmo Prophet

Time series for the Normalized Differential Vegetation Index using a short- and long-term Artificial Neural Network, and the Prophet algorithm

Séries temporais para o Índice de Vegetação Diferencial Normalizado usando uma Rede Neural Artificial de curto e longo prazo, e o algoritmo Profeta

Marco Javier Castelo Cabay ^I
m_jcastelo@yahoo.es
<https://orcid.org/0000-0003-2400-3049>

Edgar Francisco Merino Villa ^{II}
ef_merino@yahoo.es
<https://orcid.org/0000-0002-8072-1259>

Mayra Elizabeth Peñafiel Tixi ^{III}
penafiel.mayrae@gmail.com
<https://orcid.org/0000-0002-5678-5172>

Bélgica Marcela Basantes Erazo ^{IV}
marcelabasantesez@gmail.com
<https://orcid.org/0000-0001-5930-7494>

Correspondencia: m_jcastelo@yahoo.es

Ciencias Técnicas y Aplicadas
Artículo de Investigación

* **Recibido:** 23 de junio de 2022 * **Aceptado:** 12 de julio de 2022 * **Publicado:** 9 de agosto de 2022

- I. Ingeniero en Sistemas Informáticos, Máster en Tecnologías y Aplicaciones en Ingeniería Informática, Máster en TICS aplicadas a la Educación, Docente del IST Bolívar, Ecuador.
- II. Ingeniero en Sistemas Informáticos, Magíster en Tecnologías para la Gestión y Práctica Docente, Máster en Ingeniería de Software y Sistemas Informáticos, Docente IST Bolívar, Ecuador.
- III. Ingeniera en Sistemas Informáticos, Máster en Diseño y Gestión de Proyectos Tecnológicos, Docente IST Bolívar, Ecuador.
- IV. Licenciada en Ciencias de la Educación Profesora de Informática Aplicada a la Educación, Magíster en Informática Educativa, Docente IST Bolívar, Ecuador.

Resumen

La presente investigación se realizó una evaluación de modelos para el pronóstico de series temporales del Índice Normalizado de Vegetación (NDVI) por medio de una Red Neuronal Recurrente (RNR) de corto y largo plazo, y el algoritmo Prophet de Facebook.

Los datos se obtuvieron del sensor espacial Moderate Resolution Imaging Spectroradiometer (MODIS) que emite información con una periodicidad de 16 días, se obtuvieron valores desde enero de 2013 hasta diciembre del 2021 por medio de la plataforma Google Earth Engine (GEE). Utilizando el lenguaje de programación Python en un entorno Jupyter se construyó la red neuronal Long-Short Term Memory (LSTM), y el algoritmo Prophet, tomando como datos de entrenamiento 172 valores y 36 para prueba en ambos casos. Como métrica de evaluación se consideró Root Mean Square Error RMSE (RMSE) y Mean Square Error (MSE), obteniéndose valores de 0.509, 0.259 para LSTM y 0.5311, 0.2820 para Prophet, demostrando que la red LSTM tiene mejor rendimiento para la predicción de NDVI.

Palabras Clave: NDVI; series temporales; pronóstico; LSTM; Prophet.

Abstract

The present investigation was carried out an evaluation of models for the forecast of time series of the Normalized Vegetation Index (NDVI) by means of a short and long-term Recurrent Neural Network (RNR), and the Facebook Prophet algorithm.

The data was obtained from the Moderate Resolution Imaging Spectroradiometer (MODIS) space sensor that emits information with a periodicity of 16 days, values were obtained from January 2013 to December 2021 through the Google Earth Engine (GEE) platform. Using the Python programming language in a Jupyter environment, the Long-Short Term Memory (LSTM) neural network and the Prophet algorithm were built, taking 172 values as training data and 36 for testing in both cases. Root Mean Square Error RMSE (RMSE) and Mean Square Error (MSE) were considered as evaluation metrics, obtaining values of 0.509, 0.259 for LSTM and 0.5311, 0.2820 for Prophet, demonstrating that the LSTM network has better performance for NDVI prediction.

Keywords: NDVI; temporal series; forecast; LSTM; Prophet.

Resumo

A presente investigação realizou uma avaliação de modelos para a previsão de séries temporais do Índice de Vegetação Normalizado (NDVI) por meio de uma Rede Neural Recorrente (RNR) de curto e longo prazo e o algoritmo Facebook Prophet.

Os dados foram obtidos do sensor espacial Moderate Resolution Imaging Spectroradiometer (MODIS) que emite informações com periodicidade de 16 dias, valores foram obtidos de janeiro de 2013 a dezembro de 2021 por meio da plataforma Google Earth Engine (GEE). Utilizando a linguagem de programação Python em ambiente Jupyter, foi construída a rede neural Long-Short Term Memory (LSTM) e o algoritmo Prophet, tomando 172 valores como dados de treinamento e 36 para teste em ambos os casos. Root Mean Square Error RMSE (RMSE) e Mean Square Error (MSE) foram considerados como métricas de avaliação, obtendo valores de 0,509, 0,259 para LSTM e 0,5311, 0,2820 para Prophet, demonstrando que a rede LSTM tem melhor desempenho para previsão de NDVI.

Palavras-chave: NDVI; séries temporais; previsão; LSTM; Profeta.

Introducción

Las series temporales son un conjunto de datos u observaciones que hacen referencia a una o varias variables que están ordenadas cronológicamente, las series temporales se recogen en periodos de tiempo adyacentes, por esta razón pueden tener algún tipo de correlación. Han sido utilizadas en varios ámbitos como: las ciencias económicas, ciencias sociales, epidemiología, medicamentos, ciencias físicas, creando modelos usando técnicas como Modelo Autorregresivo (AR), Modelo Autorregresivo de medias móviles (ARMA), Modelo Autorregresivo Integrado de Medias Móviles (ARIMA), entre otros.

El índice de vegetación normalizado NDVI es un índice calculado por medio de las bandas Near Infrared Reflectance (NIR) y la banda en la región visible RED del espectro electromagnético que emiten las imágenes satelitales que se calcula mediante la fórmula:

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

Este índice ayuda a reconocer la presencia de vegetación dentro de un territorio específico, tipos de cobertura, daños producidos en la vegetación por fenómenos naturales o por el hombre, incendios forestales, plagas entre otros.

Se han realizado varias investigaciones utilizando NDVI para determinar el cambio de cobertura vegetal afectada por la minería (Al-Shateri, 2020; Xiao et al., 2020), determinar la fenología de distintas áreas como bosques, plantaciones, rendimiento de cultivos (Touhami et al., 2022; Pascual et al., 2022; Zhu & Lei, 2018).

La predicción de series temporales utilizando Redes Neuronales Artificiales de tipo Long Short Term Memory (LSTM), es relativamente nueva y se ha obtenido muy buenos resultados (Rodríguez, 2019; Siami-Namini et al., 2018; Omar & Kawamukai, 2022; Dubrovin et al., 2022; Wibawa et al., 2022) debido a que en las series temporales existe una secuencia de dependencia entre variables de entrada a diferencia de las regresiones normales, estas redes neuronales manejan muy bien estos parámetros de entrada ya que pueden aprender de secuencias largas y cortas de observaciones.

Prophet es un algoritmo creado por Facebook para series temporales, su principal característica es que se basa en un modelo aditivo donde las tendencias no lineales se ajustan a la estacionalidad, también tienen en cuenta el efecto de los días festivos que suelen causar valores atípicos especialmente cuando se hace predicciones para entornos comerciales.

Es un algoritmo relativamente nuevo, sin embargo, ha sido utilizado para detectar anomalías de dispositivos de Internet de las Cosas (Malki et al., 2022), pronóstico de la carga de electricidad (Shohan et al., 2022), tendencia de la producción del petróleo (Ning et al., 2022), predicción de la calidad del aire (Middya & Roy, 2022), impacto del [COVID 19](#) (Sirikonda et al., 2022).

El objetivo de este estudio consiste en hacer una comparación de una red neuronal LSTM con una serie de tiempo Propeth para pronosticar la evolución de NDVI, en zonas de cultivo de la serranía ecuatoriana.

Datos

Para el presente estudio se utilizan datos del repositorio de (Google Earth Engine) GEE; una plataforma en línea para procesar datos masivos de detección remota a gran velocidad (Gorelick et al., 2017), dentro de esta plataforma se puede encontrar el conjunto de datos Moderate-resolution Imaging Spectroradiometer-NDVI, de donde se obtuvieron las observaciones con una frecuencia de 16 días, se tomaron muestras desde enero de 2013 hasta diciembre de 2021 un total de 208 valores de NDVI con sus respectivas fechas en una zona agrícola donde anualmente se realizan

cultivos en la provincia de Chimborazo, Ecuador, específicamente en la parroquia de Punín donde se siembra papas, habas, maíz cíclicamente.

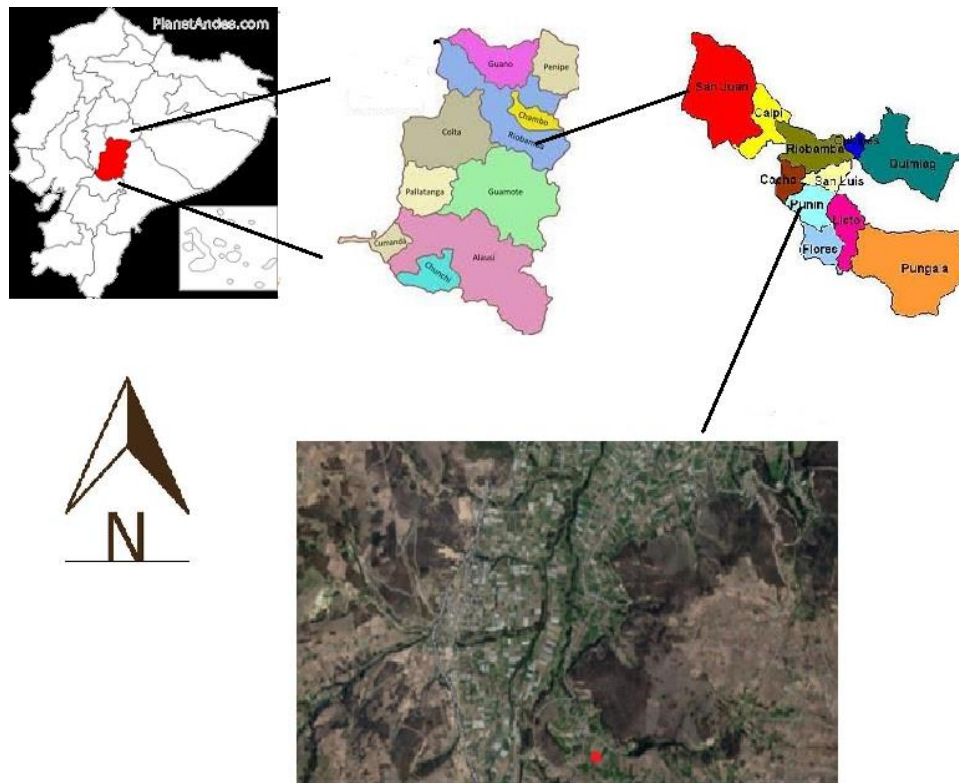


Figura 1. Sitio donde se obtuvo los datos NDVI en la Provincia de Chimborazo, Punín.

Metodología

Como herramienta se utilizó el lenguaje Python en un entorno de programación Jupiter – Anaconda, mediante su librería Tensor Flow permite crear todo tipo de RNA, también posee soporte para la implementación del algoritmo Prophet.

Red Neuronal LSTM

Las redes neuronales de tipo LSTM son un tipo especial de Redes Neuronales Recurrentes (RNR), una característica de estas redes es que pueden recordar estados y utilizar esta información para poder predecir el siguiente estado, esto la transforma en una red óptima para el cálculo de series temporales; son como un perceptrón multicapa en una sola envoltura de manera que la salida del perceptrón es la entrada de la siguiente capa. (González, 2020).

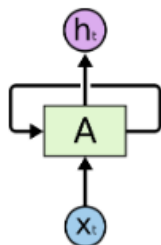


Figura 2. Representación de una Red Neuronal Recurrente.

- A es una red neuronal
- x_t es la entrada de la red
- h_t es la salida de la red

Una RNR puede ser algo como una copia de la misma red donde cada una pasa su mensaje a su sucesor, este tipo de redes ha sido ampliamente utilizado en varios escenarios como reconocimiento de voz, modelado de lenguaje, traducción, subtítulos de imágenes (Olah, 2015).

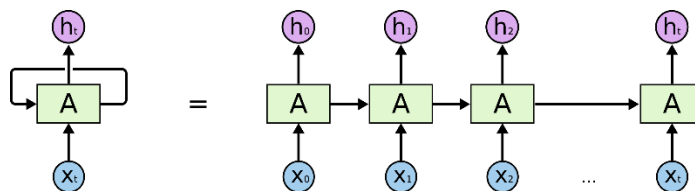


Figura 3.. Representación de una Red Neuronal Recurrente desplegada.

Toman información de su predecesor para pronosticar el siguiente paso, pero se presentan ocasiones en las que se necesita una memoria a largo plazo para poder predecir de mejor manera según el contexto, como en la predicción de palabras en donde la palabra anterior se preste para ambigüedades o confusiones.

Este tipo de redes neuronales tienen el problema de desvanecimiento de gradiente, esto quiere decir que este tipo de redes no pueden aprender de capas distantes, esto se debe a la propagación hacia atrás para recomponer los pesos en cada época de entrenamiento de la red neuronal, a medida que se alejan del punto donde empieza la retro propagación.

Este inconveniente fue resuelto por las redes LSTM (Hochreiter & Schmidhuber, 1997), tienen la capacidad de recordar a largo plazo, su estructura es similar a las RNR pero el módulo repetitivo

es distinto, en lugar de tener una sola capa de red neuronal hay cuatro que interactúan de manera especial.

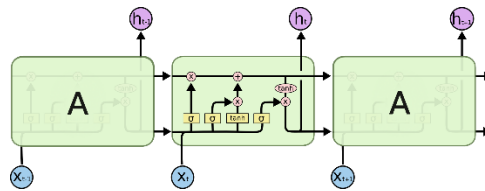


Figura 4 . Representación de una Red Neuronal Recurrente de corto y largo plazo (LSTM)

El elemento más importante de las redes LSTM es el estado de la célula, que es una línea horizontal superior que recorre todo el diagrama, el LSTM tiene la capacidad de agregar o quitar información a esta célula mediante compuertas.

La primera compuerta está formada de una capa de red neuronal sigmoidea y una operación de multiplicación. La función sigmoidea puede adquirir dos valores “0”, que no permite el paso de información y “1”, que sí permite el paso.

Posteriormente posee una capa sigmoide llamada “input gate layer” que decide qué valor se actualizará y una capa tangente hiperbólica que de acuerdo con sus propiedades crea un vector de posibles valores creados en la anterior capa que se añadirán al nuevo estado.

Finalmente existe una capa sigmoide que decide cual va a ser el “output”, seguida de otra capa tangente hiperbólica que decida que valores se van al output de la red. Por su capacidad de recordar estados anteriores en corto y largo plazo son muy utilizadas para la predicción de redes temporales (Mañas, 2019)

Algoritmo Prophet

La idea general del modelo es similar a un modelo aditivo generalizado. El Algoritmo Prophet se ajusta a la tendencia, la estacionalidad y los días festivos, tiene una alta tolerancia a los datos faltantes y a los cambios de tendencia, además de manejar bien los datos atípicos.

Tendencia: Son detectados automáticamente seleccionando distintos cambios de predisposición de la curva dentro del conjunto de datos y así obtiene una función de tendencia lineal o de crecimiento logístico.

Estacionalidad: Es modelada por medio de series de Fourier.

Feridos y fechas importantes: El usuario puede decidir si tomarlos en cuenta o no para el modelado.

Todos estos parámetros se combinan en la siguiente ecuación:

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

- $g(t)$ se refiere a la tendencia (cambios durante un largo período de tiempo)
- $s(t)$ se refiere a la estacionalidad (cambios periódicos o de corto plazo)
- $h(t)$ se refiere a los efectos de los días festivos o atípicos, esto es para cuando se realiza un pronóstico de ventas.
- $e(t)$ se refiere a los cambios incondicionales que son específicos de una empresa, una persona o una circunstancia. También se le llama término de error.
- $y(t)$ es el pronóstico.

Las principales características de Prophet son:

Es un algoritmo rápido y confiable, el usuario puede manipular los parámetros para obtener los resultados de una manera rápida y confiable. Se puede obtener los pronósticos sobre datos desordenados y faltantes con valores atípicos y fuertes cambios en las tendencias. Este algoritmo es de uso libre y está disponible en R y Python.

Pronóstico NDVI mediante redes neuronales LSTM

Una vez que se han obtenido los valores de las fechas y el NDVI respectivo se calcula la media mensual de los valores para establecer la frecuencia temporal con la que se trabajará en los ejemplos propuestos. Posteriormente se procede a graficar los valores obtenidos a través de los años como lo muestra la figura 5.

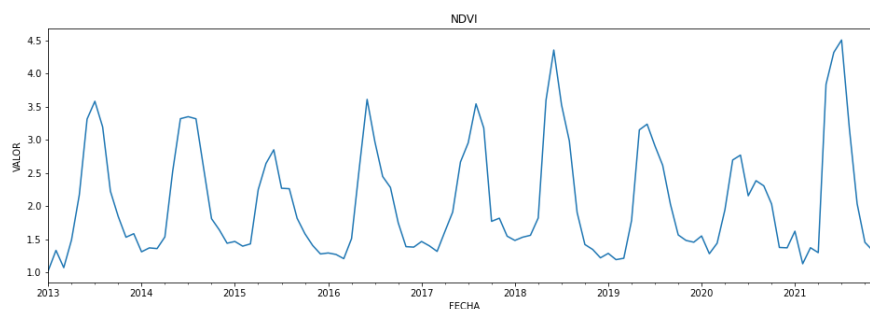


Figura 5. Gráfico del NDVI desde el año 2013 al 2021.

Posteriormente se grafica la tendencia, la estacionalidad y el ruido de esta serie temporal como se muestra en la figura 6.

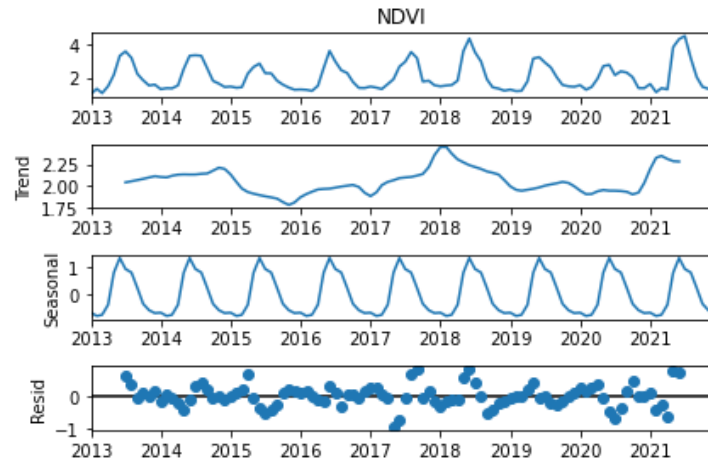


Figura 6. Tendencia, estacionalidad y ruido de la serie temporal NDVI.

De los 208 elementos disponibles se toma 172 para entrenamiento y 36 para prueba, se escala y normaliza tanto los datos de entrenamiento como de prueba para que puedan ser compatibles con los tensores de la red neuronal, después creamos un objeto generador de series temporales para transformar muestras de series temporales univariantes como multivariantes en valores listos para ser usados por modelos de aprendizaje profundo, como lote de entrada se toma un valor de 2, con un número de características de entrada de 1 ya que es una red neuronal univariante y el único parámetro de entrada son los valores de NDVI.

Posteriormente se genera la red neuronal de tipo secuencial con una capa de entrada de LSTM de 100 neuronas, una función de activación Relu, luego presenta una secuencia alternada de 100 y 200 neuronas respectivamente, con un dropout de 0.2 que evitan el sobreajuste, y finalmente una capa densa con una neurona debido a que va a predecir solo una variable continua como se muestra en la figura 7. Se utilizó el optimizador adam, y como función de pérdida Mean Squared Error (MSE) o en español el Error Cuadrático Medio, se la entrenó durante 50 épocas, dando como resultado la curva de aprendizaje.

```
[75]: from keras.models import Sequential
      from keras.layers import Dense
      from keras.layers import LSTM
      from keras.callbacks import ModelCheckpoint
      from keras.layers import Dropout

      lstm_model = Sequential()
      lstm_model.add(LSTM(100, activation='relu', return_sequences=True,
                          input_shape=(n_input, n_features)))
      lstm_model.add(LSTM(100, return_sequences=True))
      lstm_model.add(Dropout(rate=0.3))
      lstm_model.add(LSTM(200, return_sequences=True)) #se le agrega otra capa y un dropout
      lstm_model.add(Dropout(rate=0.2))
      lstm_model.add(LSTM(100, return_sequences=True)) #se le agrega otra capa y un dropout
      lstm_model.add(Dropout(rate=0.2))
      lstm_model.add(LSTM(200, return_sequences=True)) #se le agrega otra capa y un dropout
      lstm_model.add(Dropout(rate=0.2))
      lstm_model.add(LSTM(100, return_sequences=True)) #se le agrega otra capa y un dropout
      lstm_model.add(Dropout(rate=0.2))
      lstm_model.add(LSTM(100, return_sequences=False))
      lstm_model.add(Dense(1))

      mc = ModelCheckpoint('best_model_lstm.h5', monitor='val_loss', mode='min',
                          verbose=1, save_best_only=True)
```

Figura 7. Código de la Red Neuronal LSTM

En la figura 8 se observa la función de pérdida del modelo que a partir de la época 50 se estabiliza, es decir deja de aprender, entonces se detiene automáticamente utilizando una función de parada para evitar el sobreajuste.

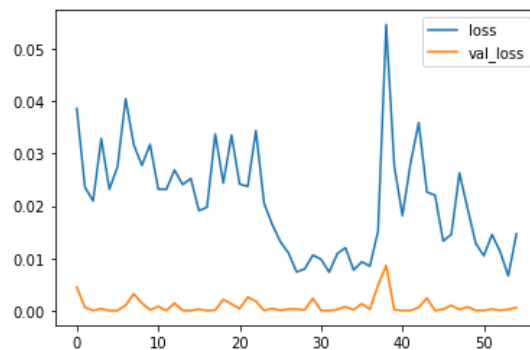


Figura 8. Gráfico de la función de pérdida del modelo LSTM

Una vez que se ha optimizado el modelo se procede a realizar las predicciones obteniendo los siguientes resultados de la figura 9, como se puede observar la curva de la predicción se ajusta bastante bien a la curva NDVI.

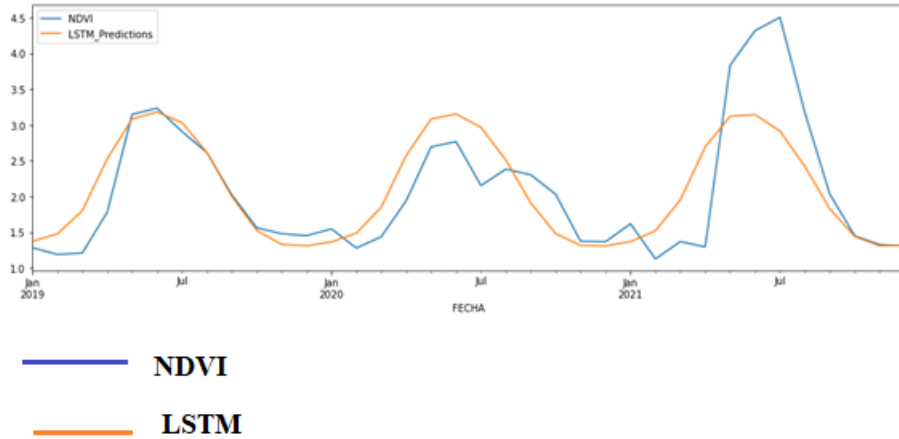


Figura 9. Curva real NDVI, y curva de Predicciones LSTM

Como resultado del ajuste de valores se obtienen las siguientes métricas de evaluación:

MSE Error: 0.25919546961969236

RMSE Error: 0.5091124331812104

Posteriormente se vuelve a entrenar la red neuronal para obtener las predicciones a partir del 01 de enero de 2019 hasta el primero de enero de 2025, en la figura 10 se muestran dos gráficos uno para el NDVI desde 2013 hasta el 2021, la figura 11 muestra la predicción desde 2022 hasta 2029.

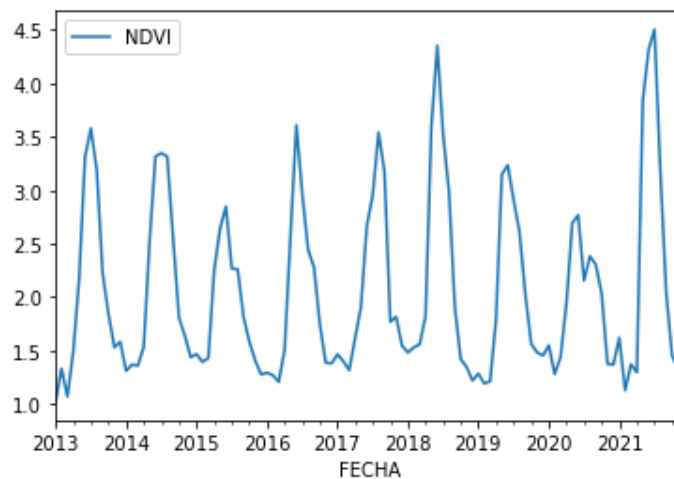


Figura 10. Curva de Predicciones LSTM 2013 a 2021

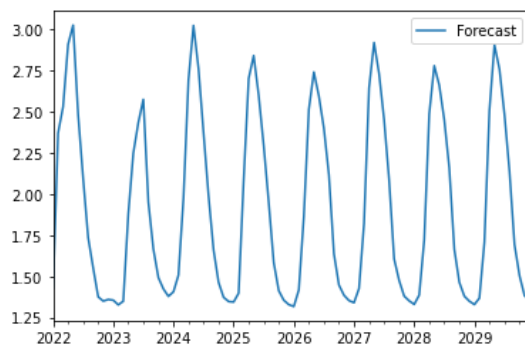


Figura 11. Curva de Predicciones LSTM 2022 a 2029

Por último se muestra la imagen sobrepuesta del NDVI con la proyección obtenida mediante este modelo ver figura 12.

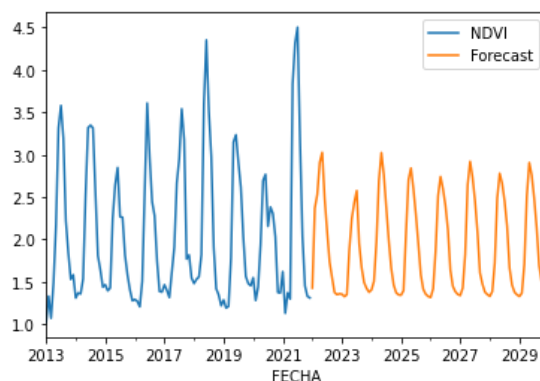


Figura 12. Curva de Predicciones LSTM 2013 - 2021 conjuntamente con las predicciones 2022 - 2029

Pronóstico de NDVI mediante el algoritmo Prophet

De igual manera que en el ejemplo anterior se toma 172 datos de entrenamiento y 36 para prueba, con una frecuencia mensual. Se debe cambiar el nombre de las variables ya que este algoritmo utiliza siempre estos mismos parámetros, a la variable de tiempo se la debe nombrar “ds”, a la variable que vamos a predecir “y”.

Posteriormente creamos el modelo y lo entrenamos por medio de la función fit(), en este caso no es necesario normalizar los datos, lo que hace de este modelo fácil de utilizar. A continuación, creamos un conjunto de datos con las predicciones con la cantidad de períodos que deseamos predecir, y la frecuencia que para este caso será en meses, posteriormente se representa el NDVI original y la serie de tiempo Prophet como se observa en la figura 13.

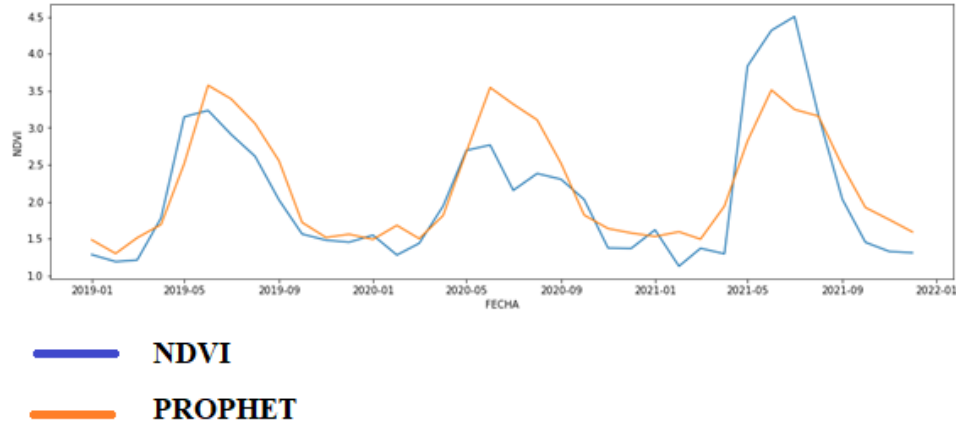


Figura 13. Serie NDVI y Prophet.

La serie temporal generada se ajusta bastante bien a la original, aprende los patrones de los picos más altos, pero le cuesta adaptarse a los picos pequeños. Las métricas de evaluación calculadas son las siguientes:

MSE Error: 0.28208022516550035

RMSE Error: 0.5311122528858663

Resultados y Discusión

La precisión y calidad del modelo se recomienda utilizar ciertas métricas como Raíz del Error Cuadrático Medio RMSE del inglés (Root Mean Squared Error), El error Absoluto Medio MAE (Mean Absolute Error), Error Cuadrático Medio MSE (Mean Squaed Error)

Cuando se trata de evaluación de modelos de regresión una de las más utilizadas es la RMSE, que es capaz de medir la precisión que existe entre dos conjuntos de datos, en el caso de series temporales permite calcular el error existente entre el valor pronosticado y el valor real.

$$\sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - y'_j)^2}$$

El MSE calcula el valor promedio entre el pronóstico y el valor real.

$$\frac{1}{n} \sum_{j=1}^n (y_j - y'_j)^2$$

Para realizar la evaluación en los dos modelos propuestos se utiliza los datos de prueba para medir el ajuste de estos datos a los valores originales de la serie temporal. A continuación, en la Tabla 1 se muestran los resultados de los dos modelos propuestos.

Modelo	Error RMSE	Error MSE
LSTM	0.509112	0.259195
Prophet	0.531112	0.282080

Tabla 1. Métricas de evaluación del modelo LSTM y Prophet

Cuanto menos sea el valor del valor de RMSE y MSE el modelo se ajusta mejor a la realidad; entonces puede realizar predicciones con mayor precisión. Por lo expuesto entonces el mejor modelo para el presente estudio son las redes neuronales de tipo LSTM, que obtiene un valor de RMSE de 0.5091 y un valor MSE de 0.2591, que es menor al valor obtenido para Prophet. Esto también se lo puede observar en el gráfico de la Figura 14 donde la línea azul muestra los valores de NDVI, la línea naranja muestra la serie generada mediante LSTM, y la línea verde representa la curva generada por Prophet.

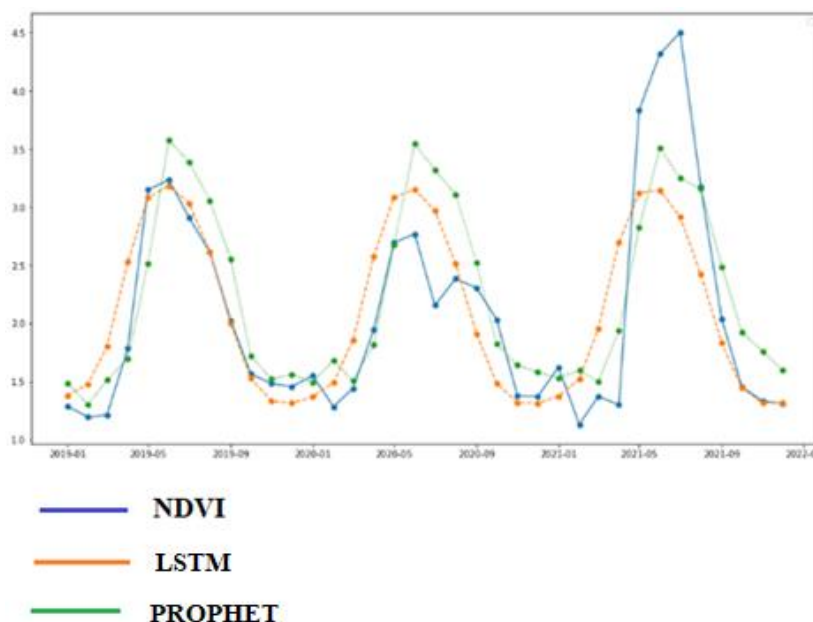


Figura 14. Serie temporal NDV original, LSTM, y Prophet

Prophet es un algoritmo que no ha dado buenos resultados en diferentes ámbitos como es el caso de la predicción de Coronavirus en una zona de Bangladesh, fue comparada con otros modelos regresivos como Regresión de Puente, Regresión Lasso, y Regresión Lineal Múltiple, obteniendo un rendimiento moderado (Rahman et al., 2022).

En otro estudio el algoritmo Prophet es fácilmente superado por una Red Neuronal LSTM, pero sin embargo supera a algoritmos tradicionales como ARIMA en el pronóstico de carga de electricidad a corto plazo (Bashir et al., 2022).

Existe una comparación entre ARIMA y Prophet (Hayashi, 2017) en donde en primera instancia se demuestra que Prophet tienen un mejor rendimiento que ARIMA, pero luego de realizar un ajuste de parámetros realizados por un usuario experto en pronósticos de series temporales, se consiguió un mejor rendimiento de ARIMA respecto a Prophet.

En el artículo controversial (Cotton, 2021) señala que el algoritmo es bueno con los pronósticos de negocios de Facebook y funciona bien dentro del entorno para el que fue creado que son los entornos comerciales y de ventas; sin embargo, no existen comparaciones con otros diversos métodos para predicción de series temporales en un ámbito diferente.

La investigación concuerda con los estudios expuestos anteriormente ya que muestra un rendimiento inferior del algoritmo Propeth a una red LSTM, al no estar en un entorno comercial.

El algoritmo Prophet tiene una gran ventaja sobre otros modelos y es la facilidad de su implementación, ya que no necesita de transformaciones en los datos de entrada para poder funcionar, por lo que sería bastante útil para usuarios y empresas que no tienen conocimientos de programación y series temporales avanzadas

El objetivo de la presente investigación no fue realizar un análisis espacio temporal del índice NDVI en un lugar predeterminado, sino más bien establecer el rendimiento de los dos algoritmos para el pronóstico de este índice, una vez determinado cuál es el mejor algoritmo se puede tomar los resultados como punto partida para realizar otro tipo de investigaciones como predecir el rendimiento de cultivos, determinar perturbaciones en la vegetación, pronosticar tendencias de la vegetación, establecer factores de perturbación de plantaciones.

Conclusiones

En el presente estudio se realizó una comparación entre una Red neuronal LSTM y el algoritmo Prophet de Facebook para establecer el mejor modelo para realizar el pronóstico de series

temporales del NDVI, obteniendo como resultado que algoritmo Prophet no supera a una red neuronal LSTM

Prophet funciona muy bien en entornos comerciales, pero en otros ambientes se ha demostrado que es superado fácilmente por distintos modelos de series temporales especialmente los creados por medio de RNA.

El índice NDVI es ampliamente utilizado sobre todo en temas ambientales así que merece ser estudiado más a fondo en su evolución espacio – temporal aplicando nuevas técnicas para series temporales mediante imágenes satelitales.

El pronóstico mediante series temporales se puede extender a otros índices obtenidos mediante bandas espectrales como EVI, BSI, NBR, NDWI, que son útiles para calcular la severidad de incendios, la expansión de las ciudades, determinar la existencia de masas de agua, utilizando los métodos presentados o variando la metodología para obtener nuevos resultados.

Referencias

1. Al-Shateri, H. A. (2020). Land use land cover change detection in the mining areas of V.D. Yalovsky coal mine-russia. *Mining Informational and Analytical Bulletin*, (6-1), 212–223. <https://doi.org/10.25018/0236-1493-2020-61-0-212-223>
2. Bashir, T., Haoyong, C., Tahir, M. F., & Liqiang, Z. (2022). Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN. *Energy Reports*, 8, 1678–1686. <https://doi.org/10.1016/j.egy.2021.12.067>
3. Cotton, P. (2021, February 3). Is Facebook's "Prophet" the time-series messiah, or just a very naughty boy? *Microprediction*. Retrieved June 23, 2022, from <https://www.microprediction.com/blog/prophet>
4. Dubrovin, K. N., Stepanov, A. S., & Aseeva, T. A. (2022). Application of lai and NDVI to model soybean yield in the regions of the Russian Far East. *IOP Conference Series: Earth and Environmental Science*, 949(1), 012030. <https://doi.org/10.1088/1755-1315/949/1/012030>
5. Gonzalez, J. M. C. (2020, April 24). Forecasting de series temporales con redes neuronales. *LinkedIn*. Retrieved June 22, 2022, from <https://www.linkedin.com/pulse/forecasting-de-series-temporales-con-redes-neuronales-casas-gonzalez/>

6. Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202, 18–27. <https://doi.org/10.1016/j.rse.2017.06.031>
7. Hayashi, H. (2017, October 18). Is prophet really better than Arima for forecasting time series data? Medium. Retrieved June 23, 2022, from <https://blog.exploratory.io/is-prophet-better-than-arima-for-forecasting-time-series-fa9ae08a5851>
8. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
9. Malki, A., Atlam, E.-S., & Gad, I. (2022). Machine learning approach of detecting anomalies and forecasting time-series of IOT devices. *Alexandria Engineering Journal*, 61(11), 8973–8986. <https://doi.org/10.1016/j.aej.2022.02.038>
10. Mañas, A. (2019). Notas sobre pronóstico del Flujo de Trafico en la Ciudad de Madrid. bookdown (Vol. I). GRIN VERLAG. Retrieved June 23, 2022, from <https://bookdown.org/amanas/traficomadrid/resumen.html>.
11. Middy, A. I., & Roy, S. (2022). Pollutant Specific Optimal Deep Learning and Statistical Model Building for Air Quality Forecasting. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4010743>
12. Ning, Y., Kazemi, H., & Tahmasebi, P. (2022). A comparative machine learning study for time series oil production forecasting: Arima, LSTM, and prophet. *Computers & Geosciences*, 164, 105126. <https://doi.org/10.1016/j.cageo.2022.105126>
13. Olah, C. (2015, August 27). Understanding LSTM networks. *Understanding LSTM Networks -- colah's blog*. Retrieved June 22, 2022, from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
14. Omar, M. S., & Kawamukai, H. (2022). Evaluation of stochastic and artificial neural network models for multi-step lead forecasting of Ndvi. *IOP Conference Series: Earth and Environmental Science*, 1008(1), 012014. <https://doi.org/10.1088/1755-1315/1008/1/012014>
15. Pascual, A., Tupinambá-Simões, F., Guerra-Hernández, J., & Bravo, F. (2022). High-resolution planet satellite imagery and multi-temporal surveys to predict risk of tree mortality in tropical eucalypt forestry. *Journal of Environmental Management*, 310, 114804. <https://doi.org/10.1016/j.jenvman.2022.114804>

16. Rahman, M., Ahmed, M. T., Nur, S., & Touhidul Islam, A. Z. (2022). The prediction of Coronavirus Disease 2019 outbreak on Bangladesh Perspective Using Machine Learning: A Comparative Study. *International Journal of Electrical and Computer Engineering (IJECE)*, 12(4), 4276. <https://doi.org/10.11591/ijece.v12i4.pp4276-4287>
17. Rodríguez, A. A. (2019). Análisis de las series temporales a la luz de Deep Learning. *Anuario Jurídico y Económico Escorialense*, II, 257–276.
18. Shohan, M. J., Faruque, M. O., & Foo, S. Y. (2022). Forecasting of Electric Load using a hybrid LSTM-neural prophet model. *Energies*, 15(6), 2158. <https://doi.org/10.3390/en15062158>
19. Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A comparison of Arima and LSTM in forecasting time series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). <https://doi.org/10.1109/icmla.2018.00227>
20. Sirikonda, S., Kumar, S. N., Sravanthi, T., Srinivas, J., Manchikatla, S. T., & Kumaraswamy, E. (2022). Forecast the death and recovery rate of COVID 2019 using Arima and Prophet Models. *INTERNATIONAL CONFERENCE ON RESEARCH IN SCIENCES, ENGINEERING & TECHNOLOGY*. <https://doi.org/10.1063/5.0081771>
21. Touhami, I., Moutahir, H., Assoul, D., Bergaoui, K., Aouinti, H., Bellot, J., & Andreu, J. M. (2022). Multi-year monitoring land surface phenology in relation to climatic variables using Modis-ndvi time-series in Mediterranean Forest, Northeast Tunisia. *Acta Oecologica*, 114, 103804. <https://doi.org/10.1016/j.actao.2021.103804>
22. Wibawa, A. P., Utama, A. B., Elmunsyah, H., Pujianto, U., Dwiyanto, F. A., & Hernandez, L. (2022). Time-series analysis with smoothed convolutional neural network. *Journal of Big Data*, 9(1). <https://doi.org/10.1186/s40537-022-00599-y>
23. Xiao, W., Deng, X., He, T., & Chen, W. (2020). Mapping annual land disturbance and reclamation in a surface coal mining region using Google Earth engine and the LANDTREND algorithm: A case study of the shengli coalfield in Inner Mongolia, China. *Remote Sensing*, 12(10), 1612. <https://doi.org/10.3390/rs12101612>
24. Zhu, W., & Lei, H. (2018). Urban Vegetation Coverage Monitoring Technology based on NDVI. *Proceedings of the 2018 7th International Conference on Energy, Environment and Sustainable Development (ICEESD 2018)*. <https://doi.org/10.2991/iceesd-18.2018.291>

© 2022 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).