

# Modelo ARIMA-ANFIS-p-convexo para series de tiempo

Manuel Alejandro Goyo

29 de agosto de 2021

## Resumen

En el siguiente trabajo introducimos un novedoso modelo híbrido llamado ARIMA-ANFIS-p-convexo el cual resulta de la combinación p-convexa de un modelo ARIMA tradicional con un modelo ANFIS. El modelo es entrenado con una serie de tiempo estacionaria y con una no estacionaria, es comparado los resultados obtenidos con un ARIMA tradicional y un modelo ANN con las métricas mean squared error, mean absolute error y mean percentage error, mostrando una mejora significativa en cada uno de las métricas.

## 1. Introducción

Las series de tiempo juegan un papel importante ya que estas permiten recopilar información de distintos fenómenos, tanto en las finanzas, como en el suministro de energía, precio de las acciones y la atención médica, por este motivo existe un gran interés como campo de estudio lo relevante a la predicción de series de tiempo o al pronóstico de valores futuros basados en los datos observados previamente.

Ante esta problemática existen varias técnicas de soft computing para el estudio de series de tiempos. Uno de estos son las redes neuronales artificiales (ANN) que son esencialmente estimadores de regresión semiparamétricos y son muy adecuados para este propósito, ya que pueden aproximarse virtualmente a cualquier función (medible) hasta un grado arbitrario de precisión [HK89]. Una ventaja significativa del enfoque ANN en el modelado de sistemas es que no es necesario tener una relación física bien definida para convertir sistemáticamente una entrada en una salida. Más bien, todo lo que se necesita para la mayoría de las redes es una colección de ejemplos representativos (pares de entrada-salida) del mapeo deseado. Luego, la ANN se adapta para reproducir la salida deseada cuando se le presenta una entrada de muestra de entrenamiento. La aparición de la tecnología de redes neuronales ha proporcionado muchos resultados prometedores en el campo de la simulación.

Por otra parte está el enfoque basado en reglas difusas. Introducido por primera vez por Zadeh [Zad65], la lógica difusa y la teoría de conjuntos difusos se emplean para describir el pensamiento y el razonamiento humanos en un marco matemático. El modelado basado en reglas difusas es un esquema de modelado cualitativo en el que el comportamiento del sistema se describe utilizando un lenguaje natural [JS93]

En los últimos años, la integración de las redes neuronales y la lógica difusa ha dado lugar a nuevas investigaciones sobre sistemas neuro-difusos. Los sistemas neuro-difusos tienen el potencial de capturar los beneficios de estos dos campos en un solo marco. Los sistemas neuro-difusos eliminan el problema básico en el diseño de sistemas difusos (la obtención de un conjunto de reglas difusas si-entonces) mediante el uso efectivo de la capacidad de aprendizaje de una ANN para el perfeccionamiento de las reglas difusas si-entonces y la optimización de parámetros. Como resultado, esos sistemas pueden utilizar información lingüística del experto humano, así como datos medidos durante el modelado.

Otro conjunto de modelos importante a considerar cuando se trata de predicciones de series de tiempo son aquellos autoregresivos (AR), Moving Average (MA), Autoregressive Moving Average (ARMA), y Autoregressive Integrated Moving Average (ARIMA). Estos modelos tratan de establecer relación entre el fenómeno sin tener una descripción interna del proceso físico que lo determina. Este tipo de modelo son clasificados como Autoregressive Integrated Moving Average (ARIMA) models [BG70].

## 2. Modelos híbridos ARIMA-ANFIS

En los últimos años se han desarrollado muchos modelos híbridos del tipo ANFIS de manera satisfactoria. Azadeh et al. [Z09] propusieron un modelo híbrido ANFIS para hacer pronósticos de series de tiempo mensuales de la electricidad y obtuvieron buenos resultados. Li et al. [LK11] compararon ANN y un modelo híbrido genetic-ANFIS para pronosticar la electricidad diaria requerida por un hotel. Li and Hu [LC12] propusieron un modelo ARIMA-fuzzy para pronósticos de series de tiempo. Se aplicaba un modelo Sugeno fuzzy en una data input-output para producir las reglas fuzzy, luego, un modelo ARIMA. Babu and Reddy [BC14] propusieron un modelo híbrido de ARIMA y ANN basado en el filter de moving-average. Suhartono et al. [SI12a] presentaron un método de ensamble basado en ANFIS-ARIMA. Lohani et al. [LA12] compararon técnicas autoregresivas, ANN y ANFIS para modelar series de tiempo. Suhartono et al. [SI12b] propusieron un modelo seasonal de dos niveles basado en el ANFIS-ARIMA. Cheng et al. [CC13] presentaron un promedio ordenado y ponderado (OWA) basado en el modelo ANFIS. Eğrioglu et al. [EE14] propusieron una modificación de ANFIS que llamaron MANFIS para el pronóstico de series de tiempo. Busenur Sarica [BS16] propusieron un modelo híbrido para la predicción de series de tiempo llamado AR-ANFIS.

## 3. El método propuesto ARIMA-ANFIS-p-convexo para series de tiempo

### 3.1. ARIMA

Los procesos autorregresivos de medias móviles ARMA(p,q) determinan a  $Y_t$  en función de su pasado hasta el retardo  $p$ , de la innovación contemporánea y el pasado de la innovación hasta el retardo  $q$ :

$$Y_t = \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + a_t + \theta_1 a_{t-1} + \dots + \theta_q a_{t-q}$$

Donde  $a_t$  es un ruido blanco con media cero y varianza  $\sigma^2$ . Este modelo se puede escribir en términos del operador de retardos como sigue:

$$(1 - \phi_1 L - \dots - \phi_p L^p) Y_t = (1 - \theta_1 L - \dots - \theta_q L^q) a_t$$

$$\Phi(L) Y_t = \Theta_q(L) a_t$$

. Además, si una serie  $Y_t$  no es estacionaria, pero su diferencia de orden  $d$ ,  $\Delta^d Y_t$  es estacionaria, se determina el modelo Autorregresivo Integrado de Medias Móviles de orden  $(p, d, q)$  o ARIMA( $p, d, q$ ), donde  $p$  es el orden del polinomio autorregresivo estacionario,  $d$  es el orden de integración de la serie, es decir, el número de diferencias que hay que tomar a la serie para que sea estacionaria, y  $q$  es el orden del polinomio de medias móviles como sigue :

$$\Phi(L) \Delta^d Y_t = \Theta_q(L) a_t$$

### 3.2. ANFIS

Supongamos que tenemos  $X = (x_1, x_2, \dots, x_n) \in R^n$  y  $Y \in R$ . Las  $J$  reglas de tipo Takagi y Sugeno son de la siguiente forma: si  $x_n$  es  $\mu_{1j}(x_1)$ ,  $x_2$  es  $\mu_{nj}(x_n)$ , ...,  $x_n$  es  $\mu_{nj}(x_n)$  entonces

$$f_j(X) = b_{0j} + b_{1j}x_j + b_{2j}x_j + \dots + b_{nj}x_j$$

donde  $\mu_{ij} = e^{-((x_i - m_{ij})/\sigma_{ij})^2}$  la cual se conoce como la función de membresía del conjunto asociado con  $x_i$ . El conjunto de parámetros  $(m_{ij}, \sigma_{ij})$  se les conoce como parámetros antecedentes y  $b_{0j}, b_{1j}, \dots, b_{nj}$  los parámetros consecuentes. El diagrama del modelo lo podemos ver en la figura 1 que vamos a estudiar detalle a detalle.

- **Capa 1:** En esta capa todos los nodos son nodos adaptativos y los resultados de esta capa son el grado de membresía difusa de las entradas, que están dadas por las siguientes ecuaciones:

$$O_{i,k} = \mu_{A_i^k}(x_i), i = 1, 2, \dots, n, k = 1, 2, \dots, J$$

donde  $x_i$  son las entradas del nodo  $i$ , y  $A_i^k$  son las etiquetas lingüísticas asociadas con esta función de nodo,  $\mu_{A_i^k}(x_i)$  pueden adoptar cualquier función de membresía fuzzy (por ejemplo triangular, trapezoidal, campana, etc.)

- **Capa 2:** Los nodos son nodos fijos. Esta capa implica operadores difusos; usa el operador AND para confundir las entradas. Están etiquetados con  $\pi$ , lo que indica que funcionan como un simple multiplicador. La salida de esta capa se puede representar como:

$$O_{2,k} = \omega_k = \prod_{i=1}^n \mu_{A_i^k}(x_i)$$

- **Capa 3:** Los nodos también son nodos fijos etiquetados por N, para indicar que ellos juegan un rol de normalización a las denominadas 'fuerzas de disparo' de la capa anterior. La salida de esta capa puede ser representada como:

$$O_{3,k} = \bar{\omega}_k = \frac{\omega_k}{\sum_{k=1}^J \omega_k}$$

Las salidas de esta capa son llamadas fuerzas de disparo normalizadas.

- **Capa 4:** Los nodos son adaptativos. La salida de cada nodo es el producto entre la fuerza de disparo normalizada y un polinomio de primer orden. La salida de la capa está dada por:

$$O_{4,k} = \bar{\omega}_k f_k = \bar{\omega}_k (b_{0k} + b_{1k}x_1 + b_{2k}x_2 + \dots + b_{nk}x_n)$$

Donde  $\bar{\omega}_k$  es la salida de las Capa 3, y  $b$ 's son los respectivos parámetros.

- **Capa 5:** En esta última capa hay solo un nodo fijo. Este nodo realiza la suma de todas las señales entrantes. La producción general del modelo está dada por:

$$O_5 = \sum_k \bar{\omega}_k f_k = \frac{\sum_i \omega_k f_k}{\sum_k \omega_k}$$

Observe que el modelo anterior se puede resumir de la siguiente manera

$$y = \frac{\sum_{j=1}^J \alpha_j(x) (b_{0j} + b_{1j}x_1 + b_{2j}x_2 + \dots + b_{nj}x_n)}{\sum_{k=1}^J \alpha_k(x)}$$

$$y = \sum_{j=1}^J \frac{\alpha_j(x)}{\sum_{k=1}^J \alpha_k(x)} (b_{0j} + b_{1j}x_1 + b_{2j}x_2 + \dots + b_{nj}x_n)$$

Teniendo en cuenta que  $\alpha_j(x) = \prod_{i=1}^n \mu_{ij}(x_i)$ .

Supongamos que tenemos un conjunto de entrenamiento de  $N'$  pares  $(p_v, q_v)$ ,  $1 \leq v \leq N'$  con  $p_v = (p_{1v}, p_{2v}, \dots, p_{nv})$ . Para cada  $1 \leq v \leq N$ , tenemos que

$$y_v = \sum_{j=1}^J a_{vj} (b_{0j} + b_{1j}x_{1v} + b_{2j}x_{2v} + \dots + b_{nj}x_{nv})$$

donde  $a_{vj} = \frac{\alpha_j(p_v)}{\sum_{k=1}^J \alpha_k(p_v)}$ . Sea  $E = \frac{1}{N'} \sum_{v=1}^{N'} (q_v - y_v)^2$  el error cuadrático medio.

El problema puede ser resuelto a través del método de mínimos cuadrados de la ecuación  $AB = D$  donde

$$D = [q_1 \quad q_2 \quad \dots \quad q_{N'}]^T$$

$$A = \begin{bmatrix} a_{11} & a_{11}p_{11} & \dots & a_{11}p_{n1} & \dots & a_{1j} & a_{1j}p_{11} & \dots & a_{1j}p_{n1} \\ a_{21} & a_{21}p_{12} & \dots & a_{21}p_{n2} & \dots & a_{2j} & a_{2j}p_{12} & \dots & a_{2j}p_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N'1} & a_{N'1}p_{1N'} & \dots & a_{N'1}p_{nN'} & \dots & a_{N'j} & a_{N'j}p_{1N'} & \dots & a_{N'j}p_{nN'} \end{bmatrix}$$

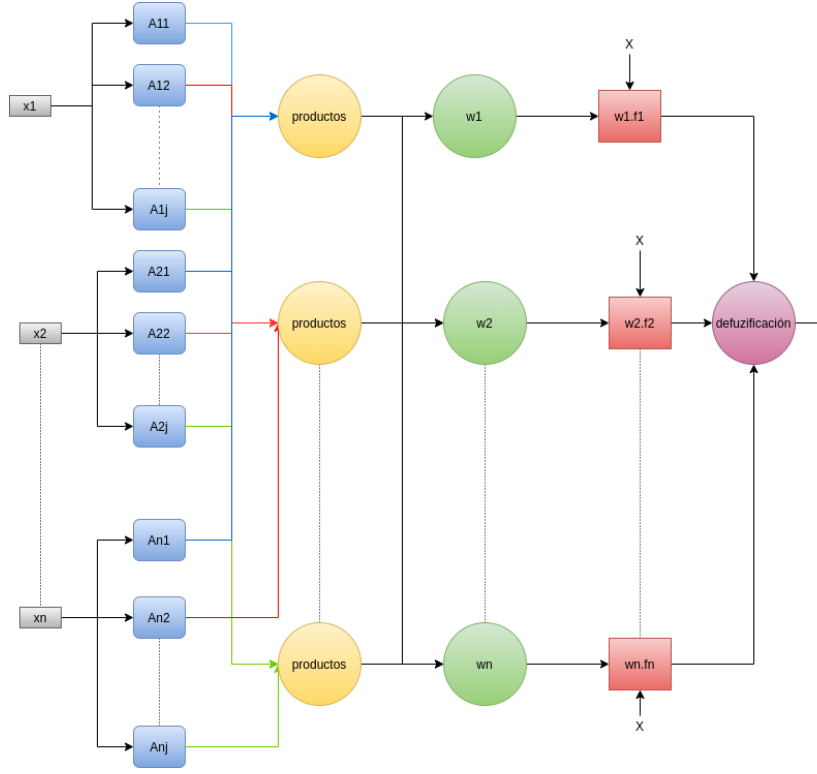


Figura 1: Diagrama del modelo ANFIS.

$$B = [b_{01} \quad b_{11} \quad \dots \quad b_{n1} \quad \dots \quad b_{0j} \quad b_{1j} \quad \dots \quad b_{nj}]^T$$

Para optimizar el valor de los parámetros antecedentes se utiliza el método del gradiente descendente de la siguiente manera:

$$m_{ij}^{new} = m_{ij}^{old} - \eta_1 \frac{\partial E}{\partial m_{ij}}$$

$$\sigma_{ij}^{new} = \sigma_{ij}^{old} - \eta_2 \frac{\partial E}{\partial \sigma_{ij}}$$

donde  $\eta_1$  y  $\eta_2$  son los hiperparámetros conocidos como learning rates.

### 3.3. ARIMA-ANFIS-p-convexo

El modelo ARIMA-ANFIS-p-convexo funciona como un modelo híbrido de la conexión entre un modelo ARIMA y un modelo ANFIS. La justificación de este modelo viene del análisis que una serie de tiempo  $y_t$  puede ser descrita de la siguiente manera

$$y_t = l_t + n_t$$

Donde la estructura de  $y_t$  contiene dos partes: una parte lineal  $l_t$  y una parte no lineal  $n_t$ , por lo que primero el pronóstico se hace de la parte lineal para luego hacer el pronóstico de la parte no lineal y combinarlas, es decir, sumar la predicción lineal y no lineal. Entendiendo este enfoque, se llevará a la práctica haciendo el pronóstico de la parte lineal con un modelo ARIMA y la parte no lineal será pronosticada por el modelo ANFIS. El resultado final de la combinación entre estos dos modelos se hará a través de una combinación  $p$ -convexa, esto es que el pronóstico será lo obtenido de sumar  $p$ -veces el modelo ARIMA mas  $(1 - p)$ -veces una transformación lineal del modelo ANFIS con  $p \geq 0$

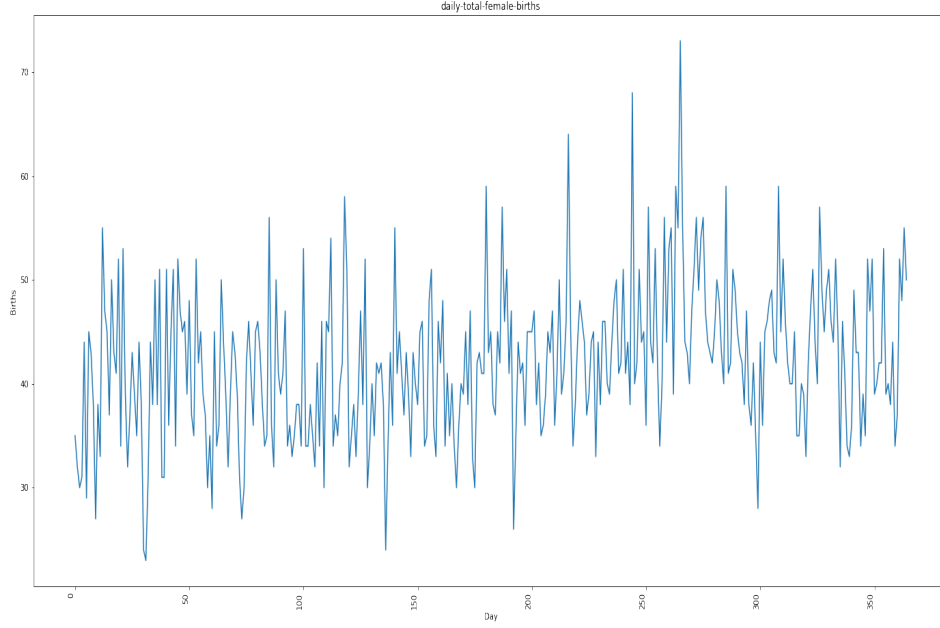


Figura 2: Diagrama del modelo ANFIS.

## 4. Aplicaciones y resultados

### 4.1. Datasets

Para poner en práctica el modelo anterior se consideran dos datasets: El primer dataset es una serie de tiempo estacionaria que registra el total diario de nacimientos femeninos, este dataset puede ser descargado gratuitamente en la url [daily-total-female-births](#) el cual podemos observar la gráfica en la siguiente figura 2.

El segundo dataset es una serie de tiempo no estacionaria que registra el total de pasajeros mensuales de una aerolínea, este dataset puede ser descargado gratuitamente en la url [airline-passengers](#) el cual podemos observar la gráfica en la siguiente figura 3

### 4.2. Métricas de rendimiento

Para comparar el rendimiento en los pronósticos hechos por cada uno de los modelos se utilizaron las siguiente métricas.

$$MAE = \frac{\sum_{i=1}^{N_t} |y_i - \hat{y}_i|}{N_t}$$

$$MSE = \frac{\sum_{i=1}^{N_t} (y_i - \hat{y}_i)^2}{N_t}$$

$$MAPE = 100 \times \frac{\sum_{i=1}^{N_t} \frac{|y_i - \hat{y}_i|}{y_i}}{N_t}$$

conocidos en la literatura como mean absolute error (MAE), root mean square error (MSE), y mean absolute percentage error (MAPE)

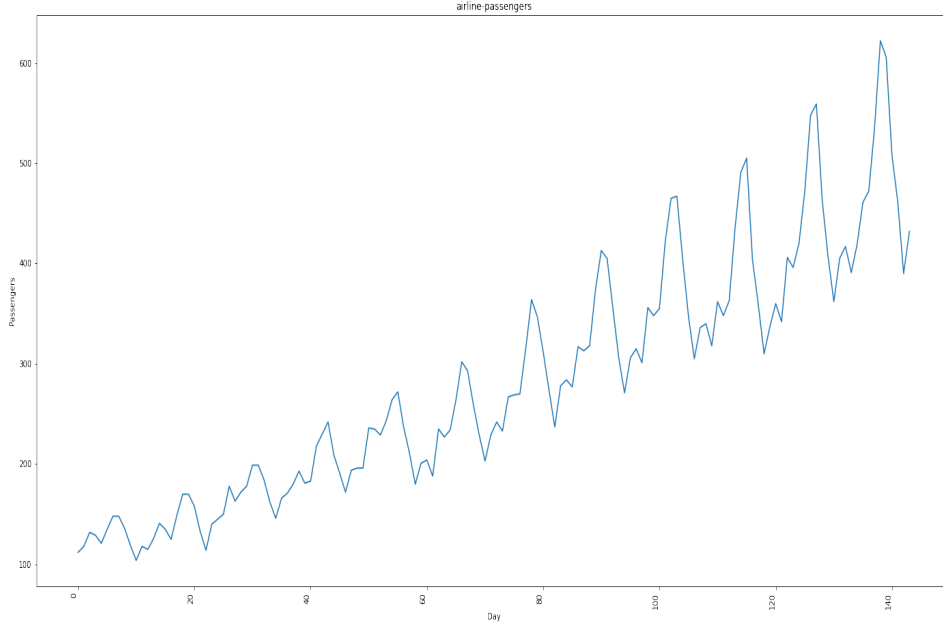


Figura 3: Diagrama del modelo ANFIS.

Modelos	MSE	MAE	MAPE
$p$ -ARIMA-ANFIS	9.80	2.56	6.16
ARIMA	38.49	4.90	11.56
ANN	34.71	4.77	11.00

Cuadro 1: An example table.

### 4.3. Resultados

Para el primer el dataset se aplicó un modelo ARIMA(2,0,2). Luego para entrenar el modelo ANFIS, se tomó una ventana de tamaño 3, esto es que se usó el valor de la serie en el tiempo  $t - 2$ ,  $t - 1$  y  $t$  para pronosticar el valor de la serie en el tiempo  $t + 1$ . Este mismo proceso se utilizó para entrenar el modelo ANN. Se determinó el valor de  $p = 0,02$  para el modelo  $p$ -convexo. Los resultados de esto lo podemos encontrar en la siguiente cuadro 1

Además podemos ver el gráfico de los pronósticos de cada uno de los modelos junto con la serie original en el siguiente gráfico 4.

Para el segundo el dataset se aplicó un modelo ARIMA(2,1,2). Luego para entrenar el modelo ANFIS, se tomó una ventana de tamaño 3, esto es que se usó el valor de la serie en el tiempo  $t - 2$ ,  $t - 1$  y  $t$  para pronosticar el valor de la serie en el tiempo  $t + 1$ . Este mismo proceso se utilizó para entrenar el modelo ANN. Se determinó el valor de  $p = 0,2$  para el modelo  $p$ -convexo. Los resultados de esto lo podemos encontrar en la siguiente cuadro 2

Además podemos ver el gráfico de los pronósticos de cada uno de los modelos junto con la serie

Modelos	MSE	MAE	MAPE
$p$ -ARIMA-ANFIS	125.87	2.56	1.77
ARIMA	13806.75	109.65	23.27
ANN	3538.87	52.55	17.10

Cuadro 2: An example table.

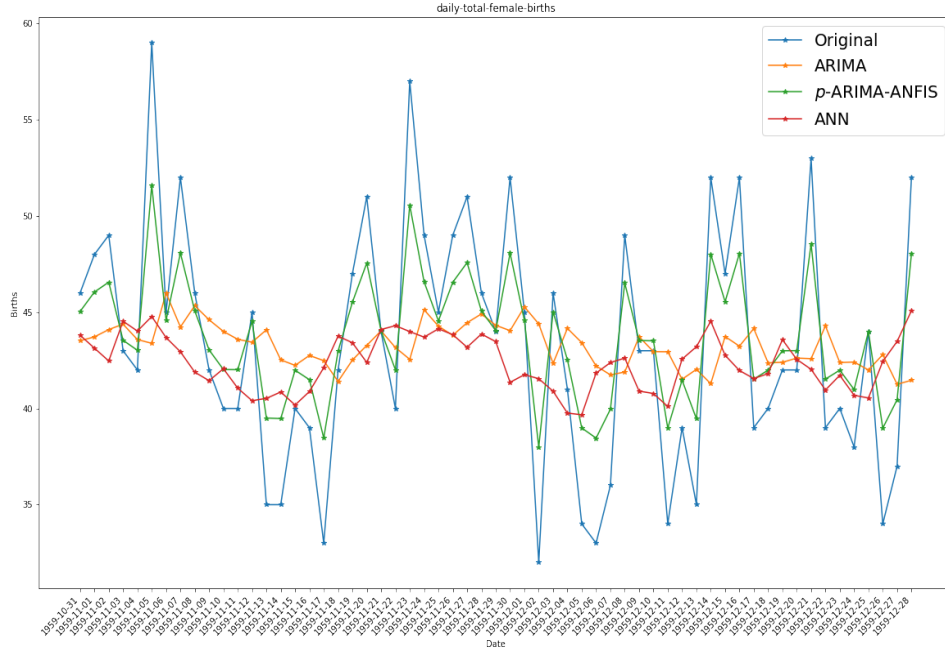


Figura 4: Pronósticos de los modelos con la serie original.

original en el siguiente gráfico 5

Se puede notar que ampliamente el modelo  $p$ -ARIMA-ANFIS es ampliamente mejor en ambos escenarios, tanto en el escenario de una serie de tiempo estacionaria, observando los resultados en la tabla 1, como en el escenario de una serie de tiempo no estacionaria, observando los resultados en la tabla 2. En cuanto a los gráficos de pronósticos, se puede ver que el modelo  $p$ -ARIMA-ANFIS también se comporta de mejor manera en ambos gráficos.

Como un dato adicional, si se desea ver de manera detallada, como fue llevado el proceso de aplicación del algoritmo, le sugerimos revisar el siguiente [link](#) que lleva a un repositorio en github con la información.

## 5. Conclusiones

Observando con detalle el estudio anterior, se puede ver que el modelo  $p$ -ARIMA-ANFIS muestra un gran desempeño en lo que a pronósticos de series de tiempo se refiere, de manera independiente si la serie de tiempo es estacionaria o no lo es, esto debido a que el modelo sabe aprovechar la composición lineal y no lineal que puede tener una serie de tiempo, combinando estos dos factores en un novedoso enfoque que se puede ajustar de una mejor manera la relación subyacente.

En futuros trabajos se puede estudiar la dependencia de este valor  $p$ , observando que si la serie tiene mayor dependencia lineal se debe tomar un  $p$  mayor, para darle mas peso a la parte ARIMA que ajusta la parte lineal del modelo. De la misma manera, cuando la serie posee mayor dependencia no lineal, se debe tomar un valor  $(1 - p)$  mayor, para darle mas peso a la parte ANFIS. También vale la pena mencionar que este tipo de modelos híbridos puede ser combinados con otros modelos del soft computing para obtener unos resultados que pueden ser mejores.

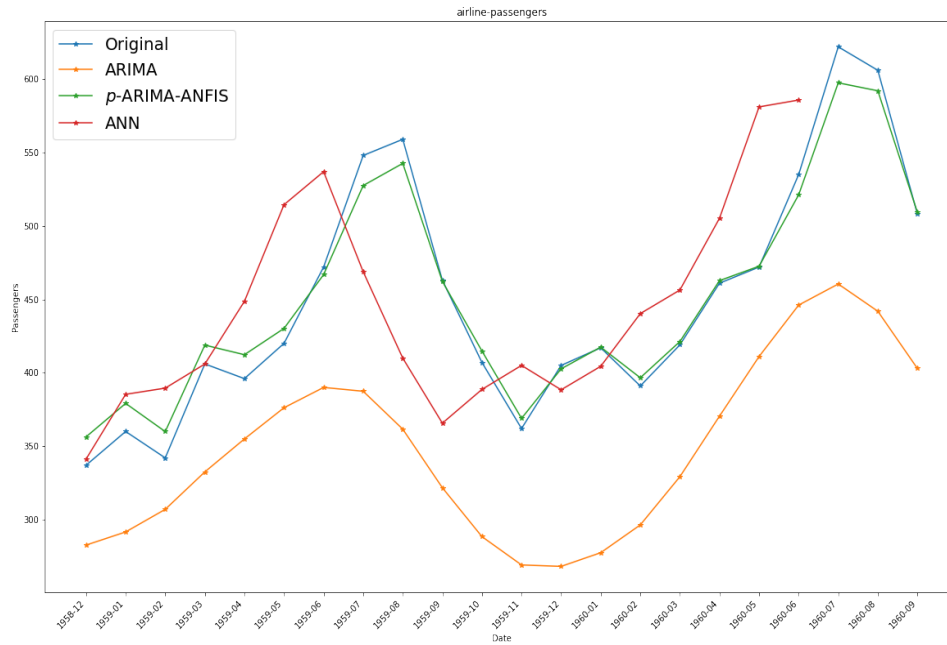


Figura 5: Pronósticos de los modelos con la serie original.

## Referencias

- [BC14] Reddy BE, Babu CN. A moving-average-filter-based hybrid arima-ann model for forecasting time series data. *Appl Soft Comput*, 2014.
- [BG70] Jenkins G.M Box G.E.P. Time series analysis: Forecasting and control. 1970.
- [BS16] Eğrioğlu E et al Busenur Sarica. A new hybrid method for time series forecasting: Ar-anfis. *Neural Comput Applic*, 2016.
- [CC13] Wei L et al Cheng C. Owa based anfis model for taiex forecasting. *Econ Model*, pages 30:442–448, 2013.
- [EE14] Aladağ H et al Eğrioğlu E. A new adaptive network based fuzzy inference system for time series forecasting. *Aloy J Soft Comput Appl*, pages 2:25–32, 2014.
- [HK89] White H Hornik K., Stichcombe M. Multi layer feed forward networks are universal approximators. *Neural Network*, (2):359–366, 1989.
- [JS93] Jang JS. Anfis: adaptive network based fuzzy inference. *IEEE*, 23(3):665–685, 1993.
- [LA12] Kumar R et al Lohani A. Hydrological time series modeling: a comparison between adaptive neuro-fuzzy, neural network and autoregressive techniques. *J Hydrol*, page 442–443:23–35, 2012.
- [LC12] Hu J-W Li C. A new arima-based neuro-fuzzy approach and swarm intelligence for time series forecasting. *Eng Appl Artif Intell*, (25):295–308, 2012.
- [LK11] Chu J. Li K, Su H. Forecasting building energy consumption using neural networks and hybrid neuro-fuzzy system: a comparative study. *Energy Build*, (43):2893–9, 2011.
- [SI12a] Faulina R et al Suhartono I. Ensemble method based on anfis-arima for rainfall prediction. *IEEE*, pages 1–4, 2012.



- [SI12b] Puspitasari M et al Suhartono I. Two-level seasonal model based on hybrid arima–anfis for forecasting short- term electricity load in indonesia. *IEEE*, pages 1–5, 2012.
- [Z09] Azadeh A. Saberi M. Gitiforouz A. Saberi Z. A hybrid simulation-adaptive network based fuzzy inference system for improvement of electricity consumption estimation. *Expert Syst Appl*, (36):11108–17, 2009.
- [Zad65] L.A. Zadeh. Fuzzy sets. *Information and Control*, (3):338–353, 1965.