



UNAH
UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

Boletín Divulgativo

Carrera de Matemáticas

- ➔ Redes Neuronales
- ➔ Criptomonedas
- ➔ Modelos estadísticos
- ➔ Teoría de Nudos
- ➔ Curvas de Rendimiento

ABRIL-2024

Presentación

Este documento fue desarrollado por el profesor David Méndez de la Carrera de Matemáticas de la UNAH, presenta artículos divulgativos y de investigación desarrollados por estudiantes del Seminario de Investigación de la Carrera de Matemáticas, curso desarrollado durante el primer período académico del año 2024. Se abarca una temática variada: redes neuronales, modelos estadísticos y sus aplicaciones, curvas elípticas y criptomonedas, curvas de rendimiento y teoría de nudos; en algunos de los trabajos se desarrolló una revisión bibliográfica de trabajos pertinentes y se resumió según lo comprendido por cada autor, en otros casos, se realizó experimentación original y se obtuvo resultados interesantes.

El objetivo principal de desarrollar este documento es que a futuro, en base a la experiencia obtenida y después de tener varias experiencias similares, se transforme en una revista científica de Matemáticas, cuestión que requiere de mucho trabajo por parte del equipo de profesores investigadores del programa y otros colaboradores externos; además de ser una muestra de que en la Carrera de Matemáticas se está desarrollando en los estudiantes un espíritu investigador. Cabe destacar que documentos similares se han desarrollado desde el programa de Maestría en Matemáticas de la UNAH.

Todas las revisiones bibliográficas y temas aquí presentados se encasillan dentro de las líneas de investigación de la UNAH, entre los temas prioritarios abarcados se encuentran: ciencia, cambio climático y vulnerabilidad, y productividad. Esto evidencia que la Carrera de Matemáticas está interesada en colaborar con las prioridades investigativas de la universidad y mantiene un compromiso con vincularse con la sociedad. Se espera que a futuro se sigan desarrollando publicaciones similares, que a pesar de ser trabajos de divulgación, forman un inicio en la actividad investigativa de la

Carrera de Matemáticas.

Abril del año 2024, Ciudad Universitaria

Tegucigalpa, M.D.C., Honduras

© Carrera de Matemáticas - UNAH

Edificio F1, Segundo Piso, Ciudad Universitaria

Tegucigalpa, M.D.C. Honduras.

[https://matematica.unah.edu.hn/escuela/carreras/licenciaturas/
carrera.matematica@unah.edu.hn](https://matematica.unah.edu.hn/escuela/carreras/licenciaturas/carrera.matematica@unah.edu.hn)

Tel. 2216-3055

Contenido

1. Análisis de series de tiempo con métodos estadísticos y Machine Learning - Daniel Vásquez
..... (p. 1 - 17)
2. Feedforward Neural Networks, una explicación general de su funcionamiento y su papel en Machine Learning - Jocsán Hernández
..... (p. 18-34)
3. Invariantes polinomiales de nudos - Alan Hernández
.....(p. 35 - 54)
4. Curvas elípticas, funciones hash y su uso en criptomonedas - Jaime Sibrián
..... (p. 55-69)
5. Análisis prospectivo de la precipitación pluvial en Honduras mediante modelos estadísticos y Machine Learning - Anthony Sanchez
.....(p. 70 - 88)
6. Ajuste y calibración de curvas de rendimientos - Andrew Castejón
..... (p. 89-102)
7. Modelo de regresión múltiple aplicado a factores que causan la pobreza en Honduras - Fredys Cruz
..... (p. 103-118)

ANÁLISIS DE SERIES DE TIEMPO CON MÉTODOS ESTADÍSTICOS Y MACHINE LEARNING

DANIEL VÁSQUEZ

RESUMEN. Las series de tiempo son una herramienta muy útil para el análisis de datos, nos ayudan en la interpretación y toma de decisiones en distintos problemas donde nos encontramos con fenómenos que van variando según un determinado período de tiempo. Por ello, en este artículo nos centraremos en la estimación y predicción de series temporales, como aplicación, tomaremos una serie de datos del Banco Central de Honduras (BCH) del Índice Mensual de la Actividad Económica (IMAE) del país, con datos desde el año 2000 hasta noviembre 2023. Los métodos matemáticos por desarrollar son los modelos arima, modelos garch y redes neuronales para ajustar dicho indicador macroeconómico y hacer un análisis de los resultados de cada método.

ABSTRACT. Time series are a very useful tool for data analysis, they help us in the interpretation and decision making in different problems in which we face phenomena that vary according to a certain period of time. That is why, in this article we will focus on the estimation and prediction of time series, as an application, we will take a series of data from the Central Bank of Honduras (BCH in spanish) of the Monthly Index of Economic Activity (IMAE in spanish) of the country, with data from 2000 to november 2023. The mathematical methods to be developed are arima models, garch models and neural networks to adjust this macroeconomic indicator and to analyse the results of each method.

INTRODUCCIÓN

Los procesos estocásticos son una colección de variables aleatorias en un espacio parametral que toman valores en un espacio de estados, un caso particular de estos procesos son las series de tiempo, estas son un proceso generalmente finito donde dicho espacio parametral es el tiempo. Las series de tiempo son observaciones de los valores de una variable a lo largo de un determinado período, el cual puede ser en días, meses, años, etc. Las observaciones son equidistantes, están ordenadas y se ve influenciado por variaciones aleatorias [6].

En el estudio de series temporales es importante analizar las observaciones para detectar ciertas propiedades en el comportamiento de la serie, una de las características que se desea encontrar es como se relacionan las observaciones adyacentes en diferentes puntos del tiempo. Además, se buscan tendencias o ciclos en la serie, este tipo de análisis nos da una visión más precisa de la naturaleza de los datos que se tienen. También es importante el modelado de la serie ya que esto nos permite predecir sus valores futuros [10]. Podemos encontrar ejemplos en distintos campos

Fecha: 15 de abril de 2024.

Palabras y frases clave. series de tiempo, modelos estadísticos, redes neuronales, indicador macroeconómico.

como la ingeniería [9], física [13], meteorología [14], medicina [11, 13], economía [16, 7], etc., de fenómenos donde el estudio de series temporales es de gran relevancia.

En la economía, el estudio de series temporales es muy amplio, una importante aplicación es la estimación de indicadores macroeconómicos, estos son estadísticas que reflejan las circunstancias económicas de un país o sector y son utilizados para evaluar la situación actual y futura de la economía y mercados financieros [3]. En este trabajo indagaremos en la estimación y predicción del índice mensual de actividad económica (IMAE), este es un indicador que permite obtener señales relevantes del comportamiento de las distintas actividades económicas de un país, busca estimar la tendencia o trayectoria que pueden tener dichas actividades a corto plazo. A pesar de tener una menor cobertura que el producto interno bruto (PIB), la información que brinda muchas veces complementa la obtenida por el PIB, he aquí el valor de la predicción de este indicador macroeconómico ya que su análisis es útil para la toma de decisiones [4]. Para estimar este indicador podemos desarrollar modelos estocásticos, dinámicos, machine learning, entre otros, que se han implementado a lo largo de la historia para el análisis de series temporales.

Entre los modelos bastante utilizados en economía se encuentra el ARIMA, estos modelos permiten tratar series no estacionarias, es decir, cuando la media de las observaciones no es constante, pero que pueden llegar a ser estacionarias eliminando las tendencias cíclicas en el proceso [6, 10], funcionan bastante bien en series que presentan estacionalidad, sin embargo, los modelos ARIMA no contemplan la volatilidad que puede presentar la serie de tiempo.

Los modelos ARCH y GARCH se centran en la volatilidad del proceso, es decir una varianza no constante, lo que implica otro tipo de no estacionariedad de la serie, al modelizar la volatilidad pueden mejorar la estimación de parámetros y la precisión de predicciones [19], aunque presentan limitaciones a la hora de modelizar la rentabilidad de un activo financiero.

Las redes neuronales son una herramienta del machine learning de aprendizaje supervisado, la diferencia con otros métodos es su capacidad de aprendizaje y adaptación, logran un ajuste de datos con gran precisión por lo que se usan con frecuencia para pronósticos de series temporales ya sea que tengan un comportamiento lineal o no lineal [15]. El objetivo es implementar modelos arima, garch y redes neuronales para el análisis del IMAE de Honduras utilizando la serie de datos “Índice Mensual de Actividad Económica Base 2000=100” publicada por el Banco Central de Honduras (BCH).

1. ANTECEDENTES

El análisis de las series de tiempo se ha desarrollado a partir de la conexión de los estudios de diferentes campos científicos. En [2, 6, 10] podemos encontrar una breve historia de los principales trabajos que fueron un gran aporte en el estudio de series temporales hasta introducir los diversos modelos de ajuste de datos.

Fue el estadístico G. Yule (1927) quien propuso los modelos autoregresivos (AR) para explicar una serie temporal de manchas solares. E. Slutsky (1937) propone un modelo de medias móviles (MA) para representar los ciclos económicos, Wold en 1949 extiende este estudio de medias móviles. En los siguientes años otros estudiosos del área trabajaron arduamente en las propiedades estacionarias y no estacionarias de las series temporales, dando origen a los métodos de alisado exponencial entre

1957-1960. Más adelante, un trabajo de suma relevancia es el llamado “enfoque Box-Jenkins” (1976) quienes, estudiando la predicción y control de series temporales industriales, presentaron los modelos ARMA y ARIMA, el primero eficiente para una gran variedad de estructuras de series de tiempo estacionarias combinando los modelos AR y MA, y el segundo cubriendo esa necesidad de analizar series de tiempo no estacionarias y estacionales, específicamente cuando falta estacionariedad en la media. Box y Jenkins presentaron una clara idea para identificar el orden del modelo y la estimación de los parámetros de este. Estos modelos ARIMA por mucho tiempo se consideraron una base para el modelado de las series de tiempo [10], aún en la actualidad seguimos encontrando trabajos en áreas como la economía donde es empleado [7].

Por los años 1980 aparecen los modelos ARCH con el objetivo de estudiar de manera más profunda la volatilidad presente en las series de tiempo relacionadas con la rentabilidad de activos financieros, fue Engle (1982) quien introduce esta nueva clase de procesos donde la varianza condicionada a las observaciones pasadas no es constante, esto es lo que se conoce como volatilidad en economía y finanzas siendo un tipo de no estacionariedad bastante común en el campo. Luego Bollerslev (1986) propone los modelos GARCH generalizando la idea de Engle con resultados más concisos para predecir la varianza condicional. Ejemplos de aplicaciones de estos modelos se pueden encontrar en [18, 16].

Por otro lado, entre los años 1980 y 1990 se comienza a utilizar las redes neuronales artificiales (ANN) y las redes neuronales recurrentes (RNN) para la predicción de series temporales. Las redes neuronales pueden modelar relaciones no lineales en un conjunto de datos y son capaces de aproximar cualquier función continua, lo que las convierte en un modelo bastante flexible [9], además analizan grandes cantidades de datos. Debido a esto es que han tenido una gran evolución en los últimos años y se han convertido en una importante herramienta para predicción de series de tiempo en distintos ámbitos, podemos ver en [9, 13, 15, 17] diferentes referencias de trabajos relacionados a redes neuronales para predicción.

2. MODELADO Y PRONÓSTICO DE SERIES DE TIEMPO

2.1. Definiciones y conceptos preliminares.

Definición 2.1. Una serie de tiempo o serie temporal es una colección de observaciones $\{x_1, x_2, \dots, x_T\}$ que registran el valor de una variable de interés en determinado período de tiempo. La colección $\{x_t\}_{t=1}^T$ se suele considerar como una muestra particular de un proceso estocástico $\{x_t\}_{-\infty}^{\infty}$ heredando sus propiedades [1]. Cuando solo una variable varía con el tiempo estamos ante una serie temporal univariante, si se tienen múltiples variables se denomina serie temporal multivariante. En el análisis de series de tiempo el objetivo es extraer parámetros relevantes o características de ella para luego generar un modelo matemático adecuado que describa el proceso y poder hacer un análisis predictivo.

Definición 2.2 (Operador de retardo). Sea L el operador de retardos (también denotado como B , de backshift) definido como

$$Lx_t \equiv x_{t-1}, \quad \text{en general} \quad L^p x_t \equiv x_{t-p}$$

Es decir, es un operador que transforma una serie temporal reemplazando cada término por el término que lo precede en la serie [1].

Definición 2.3 (Operador de diferencias). Sea $\{x_t\}$ una serie temporal cualquiera, y consideremos un nuevo proceso y_t , como la diferencia de primer orden, es decir,

$$y_t = x_t - x_{t-1},$$

aplicando el operador de retardo L , tenemos lo siguiente:

$$y_t = x_t - x_{t-1} = (1 - L)x_t = \nabla x_t$$

donde $\nabla = 1 - L$, lo definimos como el operador de diferencias de primer orden. En general, $\nabla^d = (1 - L)^d$ es el operador de diferencias de orden d [1].

2.1.1. Estacionariedad. Estudiar la estacionariedad de una serie es fundamental, de hecho, para algunos autores es considerado la base del análisis de series temporales, esta propiedad supone un estado de equilibrio estadístico del proceso [8].

Definición 2.4 (Estacionariedad estricta). Un proceso estocástico es estrictamente estacionario si la distribución conjunta de $(x_{t_1}, x_{t_2}, \dots, x_{t_n})$ es idéntica a la distribución de $(x_{t_1+k}, x_{t_2+k}, \dots, x_{t_n+k}) \forall k \in \mathbb{N}$, donde k es una variación arbitraria en el tiempo, y (t_1, t_2, \dots, t_n) una colección de n enteros positivos [19].

Es decir, la distribución conjunta de $(x_{t_1}, x_{t_2}, \dots, x_{t_n})$ es invariante bajo cualquier cambio k en el tiempo. Sin embargo, cuando se tiene un conjunto de observaciones esta propiedad es difícil de verificar empíricamente, por lo que se suele considerar una versión débil de estacionariedad.

Definición 2.5 (Estacionariedad débil). Un proceso estocástico es estacionario si $E(x_t^2) < \infty \forall t \in \{1, \dots, T\}$, y

- $E(x_t) = \mu$, es constante $\forall t$,
- $Cov(x_t, x_{t+k}) = \gamma_k$, solo depende de k , $\forall t$.

Entonces, lo anterior nos indica que tanto la esperanza incondicional como la covarianza son invariantes en el tiempo. Cabe mencionar que cuando el proceso presenta una distribución normal, las definiciones son equivalentes [19].

2.1.2. Autocorrelación y función de autocorrelación (ACF).

Definición 2.6 (Autocovarianza y Autocorrelación). La covarianza entre las observaciones x_t y x_{t+k} , separada por k intervalos de tiempo, es llamada autocovarianza del retardo k de x_t , definida por

$$Cov(x_t, x_{t+k}) = E[(x_t - \mu)(x_{t+k} - \mu)] = \gamma_k,$$

tal que, $\gamma_0 = Var(x_t)$ y $\gamma_{-k} = \gamma_k$.

Cuando la dependencia lineal entre x_t y el pasado x_{t-k} es de interés, se generaliza el concepto de correlación a autocorrelación, denotado por

$$\rho_k = \frac{Cov(x_t, x_{t-k})}{\sqrt{Var(x_t)Var(x_{t-k})}} = \frac{Cov(x_t, x_{t-k})}{Var(x_t)} = \frac{\gamma_k}{\gamma_0},$$

teniendo en cuenta que $Var(x_t) = Var(x_{t-k}) = \sigma_x^2$ bajo el supuesto de estacionariedad débil.

De lo anterior se deduce que $\rho_0 = 1$, $\rho_k = \rho_{-k}$, y $|\rho_k| \leq 1$. Se dice que la serie esta correlacionada serialmente si y solo si $\rho_k = 0$ para todo $k > 0$ [1, 19].

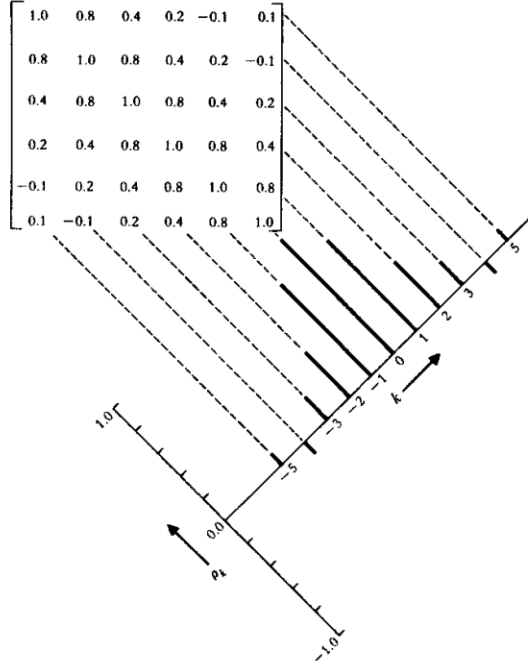


FIGURA 1. Matriz de autocorrelación correspondiente a un proceso estacionario [8].

La siguiente matriz se conoce como, matriz de covarianza asociada a un proceso estacionario (x_1, x_2, \dots, x_n) de tamaño n :

$$\begin{aligned} \Sigma_n &= \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_{n-1} \\ \gamma_1 & \gamma_0 & \gamma_1 & \cdots & \gamma_{n-2} \\ \gamma_2 & \gamma_1 & \gamma_0 & \cdots & \gamma_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_{n-1} & \gamma_{n-2} & \gamma_{n-3} & \cdots & \gamma_0 \end{bmatrix} \\ &= \sigma_x^2 \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{n-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{n-2} \\ \rho_2 & \rho_1 & 1 & \cdots & \rho_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{n-1} & \rho_{n-2} & \rho_{n-3} & \cdots & 1 \end{bmatrix} = \sigma_x^2 \mathbf{P}_n \end{aligned}$$

donde la matriz \mathbf{P}_n asociada al proceso, es la matriz de autocorrelación, ambas son simétricas definidas positivas siempre que el proceso sea estacionario. En la figura 1, podemos ver un ejemplo de esto.

Definición 2.7 (Función de autocorrelación). Al gráfico de los coeficientes de correlación ρ_k de k retardos, se le llama función de autocorrelación (ACF) $\{\rho_k\}$ del proceso estocástico. Es fácil ver que la autocovarianza se puede reescribir como $\gamma_k = \rho_k \sigma_x^2$, denotando así una función de autocovarianza $\{\gamma_k\}$ del proceso. La ACF es independiente del tamaño de la muestra que representa la serie de tiempo [8].

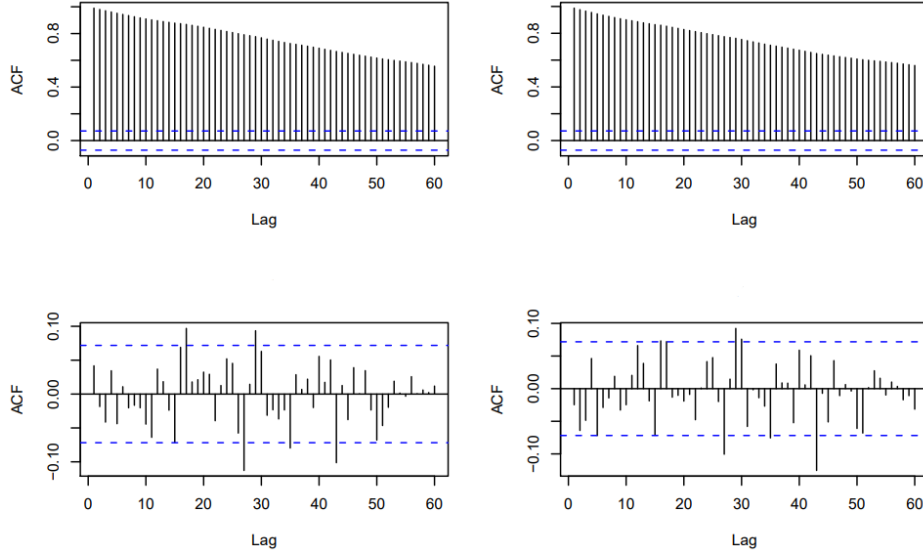


FIGURA 2. Ejemplos de funciones de autocorrelación [1].

Analizar la función de autocorrelación es importante para detectar si el proceso es estacionario y determinar que tan bien las observaciones del pasado pueden predecir el futuro.

Dada una serie de tiempo $\{x_t\}_{t=1}^T$ de T observaciones, la media de la muestra $\bar{x} = \frac{1}{T} \sum_{t=1}^T x_t$ es un estimador para $E(x_t) = \mu$, así como $s^2 = \frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})^2$ es un estimador para $Var(x_t) = \sigma_x^2$, entonces, la autocorrelación de la muestra del retardo k , está dada por:

$$\hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0} = \frac{\sum_{t=k+1}^T (x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^T (x_t - \bar{x})^2}, \quad 0 \leq k < T - 1$$

cuando $\{x_t\}$ es débilmente estacionario, $\hat{\rho}_k$ es un estimador para ρ_k [8, 19].

En el estudio de series de tiempo, se aplican algunas pruebas estadística muy importante, una de ellas es conocida como el test de Portmanteu, sirve para verificar si hay o no autocorrelación en el proceso. Contrasta la hipótesis $H_0 : \rho_1 = \rho_2 = \dots = \rho_m = 0$, contra $H_a : \rho_i \neq 0$, para algún $i \in \{1, 2, \dots, m\}$.

El estadístico propuesto es

$$Q(m) = T(T+2) \sum_{k=1}^m \frac{\hat{\rho}_k^2}{T-k},$$

una variable aleatoria chi-cuadrado con m grados de libertad.

La hipótesis H_0 se rechaza si $Q(m) > \chi_\alpha^2$, donde χ_α^2 es el percentil $100(1-\alpha)$ de una distribución chi-cuadrado con m grados de libertad. El valor m se selecciona aproximando $m \approx \ln(T)$, con T el número de observaciones de la serie [19].

Además, existe una prueba que indica si existe o no raíz unitaria en la serie temporal, esto nos dice si la serie es no estacionaria, a su vez nos indica si la serie de tiempo necesita ser diferenciada para obtener un nuevo proceso estacionario. El

termino raíz unitaria hace referencia al coeficiente que acompaña a la componente autorregresiva del modelo. Realizar esta prueba es necesario, ya que las estimaciones de parámetros y los estadísticos de pruebas desarrolladas para series temporales estacionarias no se aplican cuando una raíz unitaria está presente en el modelo.

La prueba es conocida como test de raíz unitaria aumentado de Dickey-Fuller (ADF), contrasta la hipótesis $H_0: \beta = 1$ contra $H_a: \beta < 1$, aplicando la regresión

$$(2.1) \quad x_t = c_t + \beta x_{t-1} + \sum_{i=1}^{p-1} \phi_i \Delta x_{t-i} + e_t,$$

donde c_t es una función determinista del tiempo t , $\Delta x_j = x_j - x_{j-1}$ es la serie de diferencias de x_t , e_t es el término de error, el valor p se fija de tal forma que el e_t sea correlacionado serialmente. El ADF se basa en las estimaciones por mínimos cuadrados de (2.1), el estadístico \hat{t} está dado por

$$\text{ADF-test} = \hat{t} = \frac{\hat{\beta} - 1}{\text{std}(\hat{\beta})},$$

donde $\hat{\beta}$ es el valor estimado por mínimos cuadrados de β . La hipótesis nula, $\beta = 1$ es rechazada cuando \hat{t} es suficientemente negativo [8, 19].

2.1.3. Ruido blanco y descomposición de una serie. En el análisis de series de tiempo, el teorema de descomposición de Wold nos permite entender como se combinan los aspectos determinista y estocástico en la serie observada.

Definición 2.8 (Ruido blanco). Un proceso o serie de tiempo $\{\epsilon_t\}$ se denomina ruido blanco si es una sucesión de variables aleatorias independientes e idénticamente distribuidas con $E(\epsilon_t) = 0$ y $\text{Var}(\epsilon_t) = \sigma_\epsilon^2 < \infty$, tal que, su ACF son todas cero, es decir, no están correlacionadas serialmente. En particular, si su distribución es normal, se denomina ruido blanco gaussiano.

Teorema 2.9 (Teorema de descomposición de Wold). *Todo proceso débilmente estacionario x_t , se puede expresar como combinación lineal de una sucesión de variables aleatorias no correlacionadas, escrita como*

$$x_t = \mu_x + \sum_{i=0}^{\infty} \eta_i \epsilon_{t-i},$$

donde μ_x es la media de x_t , $\eta_0 = 1$ y $\{\epsilon_t\}$ es un proceso de ruido blanco.

La variable aleatoria ϵ_t denota la información en el instante t del proceso y se le suele llamar innovación o choque en el tiempo t .

2.1.4. Volatilidad. En finanzas, es de interés cuando el precio de un activo disminuye y aumenta rápidamente en cortos períodos de tiempo. A esto se le conoce como volatilidad:

Definición 2.10 (Volatilidad). Es una medida de variación o dispersión de una variable de interés a lo largo del tiempo, y se define como la varianza condicional de la serie.

Cuando el precio del activo varía regularmente en plazos cortos, se dice que tiene alta volatilidad, si el precio se mantiene casi constante entonces tiene baja volatilidad, estos períodos de variabilidad indican heterocedasticidad en el proceso

[16]. Observar la volatilidad no es posible de forma directa, sin embargo, tiene las siguientes características:

- Existen agrupamientos de volatilidad.
- Evolucionan con el tiempo de manera continua.
- No diverge al infinito, se puede decir que la volatilidad es estacionaria.
- Parece reaccionar de manera diferente a un gran aumento de precios o una gran caída de precios [19].

2.1.5. Deep Learning. Este es un subcampo de la inteligencia artificial y el machine learning que se centra en la creación de modelos de redes neuronales profundas capaces de extraer patrones de grandes conjuntos de datos complejos obteniendo resultados precisos para la toma de decisiones [12].

Definición 2.11 (Red neuronal artificial). Es un modelo computacional inspirado en la estructura del cerebro humano, consiste en unidades sencillas de procesamiento de información, llamadas neuronas. La red neuronal está compuesta por capas en las que interactúan múltiples neuronas quienes se encargan de modelizar relaciones complejas. La salida de la i -ésima neurona está dada por:

$$y_i = \varphi \left(\sum_{j=1}^n x_j w_{i,j} + b_i \right)$$

donde φ se le conoce como función de activación, x_j son las entradas, $w_{i,j}$ los pesos asignados a cada entrada y b_i es un término de sesgo [13, 12].

Definición 2.12 (Función de Activación). La función de activación es quien define la salida de una neurona a partir de la suma ponderada de las entradas de la neurona. Es la función de activación la que da una arquitectura no lineal a la red neuronal permitiendo el mapeo de estructuras muy complejas. La función de activación lineal, sigmoide, tanh, ReLU, son ejemplos de funciones muy utilizadas en las redes neuronales [12].

Las definiciones y conceptos previos son fundamentales en el estudio de series de tiempo y para comprender algunos de los métodos que se utilizan para el modelado de las mismas. Con esto, podemos dar paso a la exploración de los datos y los respectivos métodos matemáticos que serán implementados para estimar y pronosticar el indicador macroeconómico IMAE. Para la implementación de los modelos se utilizarán los lenguajes de programación R para el análisis estadístico y Python para el enfoque de machine learning.

2.2. Descripción de los Datos.

2.2.1. IMAE. El índice mensual de actividad económica es un indicador de volumen de producción, cuya función es adelantar la tendencia o trayectoria que podría tener la producción del país, busca mostrar que ocurre con los sectores económicos y cómo evolucionan a corto plazo, refleja las variaciones de cantidad de una muestra representativa de actividades económicas del país sin considerar los costos de producción, lo que lo hace un indicador con menor cobertura que el producto interno bruto, pero con información útil para la formulación de políticas económicas [3, 4].

La fórmula utilizada para calcular el IMAE es del tipo Laspeyres,

$$\text{Índice} = \frac{\sum p_0 q_{i,t}}{\sum p_0 q_0} = \sum \left(\frac{q_{i,t}}{q_0} * W_{i,0} * 100 \right),$$

con $W_{i,0} = \frac{VA_{i,0}}{PIB}$

donde,

- p_0 : precio básico de la actividad i en el año base.
- $q_{i,t}$: volumen de producción de la actividad i en el mes t .
- q_0 : volumen de producción promedio mensual en el año base.
- $W_{i,0}$: porcentaje que representa el valor agregado del bien i respecto al PIB en el período base 2000.
- VA : valor agregado de la actividad i en el año base [3].

2.2.2. *Datos.* La base de datos utilizada en este trabajo, “Serie Índice Mensual de Actividad Económica Base 2000=100”, son datos proporcionados por el BCH por medio de su sitio web oficial [5], estos incluyen medidas absolutas y relativas de indicadores de producción que describen el comportamiento de cada actividad económica que son parte de la muestra representativa para el cálculo del IMAE de Honduras. La base de datos contiene series por actividad económica, serie del índice original, serie variación acumulada que indica en porcentaje la variación acumulada desde el comienzo del año hasta el período bajo análisis, serie variación interanual mostrando en porcentaje la comparación de un período con el mismo período del año anterior, y serie de tendencia ciclo, todo esto con frecuencia mensual desde enero del año 2000 hasta noviembre del 2023.

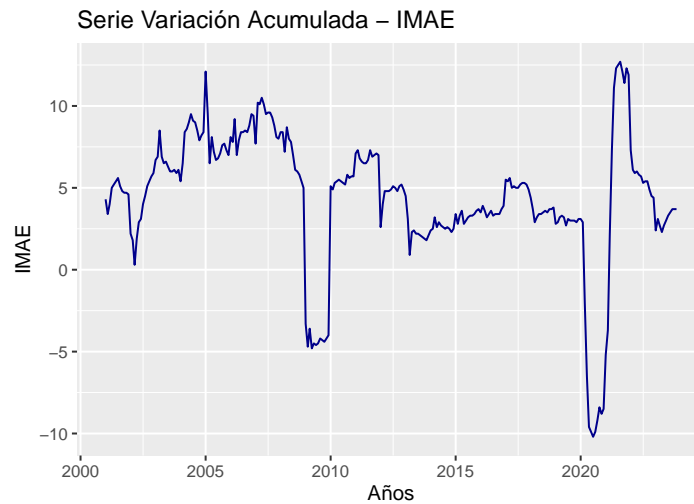


FIGURA 3. Serie temporal variación acumulada del IMAE

Los resultados del IMAE son muy variables, esto debido a que recoge el efecto de algunos comportamientos atípicos, por causas económicas, factores estacionales, climatológicos, días trabajados en cada mes, huelgas, incremento en precios de

materia prima, etc [3]. En la figura 3, se pueden apreciar algunos picos con mayor impacto de este tipo de comportamientos, como el período del año 2009, donde hubo una caída económica en gran medida por la crisis financiera mundial conocida como la “gran recesión”, además del conflicto político causado por el golpe de estado en Honduras. El período entre marzo 2020 a marzo 2021, donde las medidas de aislamiento establecidas por la pandemia del covid-19 causaron una parálisis en la economía hondureña decayendo gravemente.

2.3. Modelo autorregresivo integrado de media móvil - ARIMA. Un modelo ARIMA de orden p , d y q , ó ARIMA(p, d, q), se puede representar como sigue:

$$(2.2) \quad \phi(L)\nabla^d x_t = \mu + \theta(B)\epsilon_t,$$

donde, μ es la media (constante) de x_t , $\{\epsilon_t\}$ es un ruido blanco, $\phi(L)$ y $\theta(B)$ tienen la forma:

$$\begin{aligned} \phi(L) &= 1 - \phi_1 L - \phi_2 L^2 - \dots - \phi_p L^p \\ \theta(B) &= 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q, \end{aligned}$$

son los operadores polinomiales de los retardos que corresponden a la componente autorregresiva (AR) y media móvil (MA), respectivamente. El orden de estos polinomios lo dictan los parámetros p y q , indicando la cantidad de retardos a considerar para modelar nuestros datos. El parámetro d , corresponde a la componente de integración (I), que nos indica el orden de las diferencias que debemos tomar para hacer de nuestra serie original un proceso estacionario [1].

Los modelos ARIMA predicen valores futuros considerando solamente el pasado de la serie de tiempo, sin identificar factores subyacentes que pueden influir en el comportamiento de la variable de interés [7].

Procederemos a realizar el análisis estimando la serie variación acumulada de la base de datos IMAE.

2.3.1. Identificación de los parámetros. Para determinar los parámetros p, d, q , en primer lugar, es necesario determinar si la serie presenta estacionariedad o no. Al observar la gráfica de los datos (figura 3) se puede intuir que no hay estacionariedad, para verificarlo aplicamos el test de Dickey-Fuller que nos confirma que necesitamos aplicar las diferencias de primer orden, obteniendo así datos estacionarios, es decir, $d = 1$.

Para la elección de los parámetros p y q puede resultar útil (en algunos casos) observar las funciones de autocorrelación (ACF) y autocorrelación parcial (PACF), en este caso, graficamos la ACF y PACF de la serie ya estacionaria, y observamos que los rezagos 1, 2, 3 y 4 son los significativos, siendo estos los posibles valores para dichos parámetros (figura 4).

2.3.2. Estimación del modelo. Debido a que la elección de los parámetros p y q suele ser algo subjetivo, se experimentó con las posibles combinaciones de estos parámetros seleccionados anteriormente para especificar el modelo ARIMA. Se realizaron pruebas de estimación de los coeficientes y la prueba de Portmanteu a los residuos para comprobar que estos sean ruido blanco, es decir, residuos independientes, lo que indica un buen ajuste.

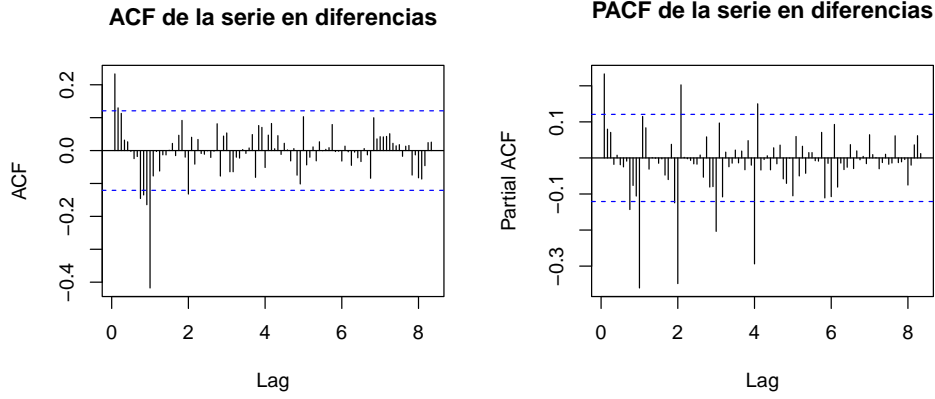


FIGURA 4. Correlograma de la serie en primeras diferencias.

Modelo	AIC	MSE	MAE
ARIMA(0,1,1)	886.0062	1.668299	0.6447181
ARIMA(1,1,0)	883.8348	1.654496	0.6450497
ARIMA(1,1,1)	883.4431	1.639396	0.6469845
ARIMA(2,1,2)	870.8342	1.529304	0.6565016
ARIMA(3,1,3)	872.4085	1.495578	0.6729038
ARIMA(4,1,4)	857.6264	1.382126	0.6721025

TABLA 1. Capacidad predictiva de los modelos ARIMA.

El criterio de información de Akaike (AIC) es una medida de la calidad relativa del modelo, el preferido es aquel que tiene un menor valor de AIC. El error cuadrático medio (MSE) y error medio absoluto (MAE) nos proporcionan una visión de la capacidad predictiva del modelo, siendo más adecuado el que tenga menor valor [7].

La tabla 1 muestra los modelos que presentan el mejor ajuste, entre estos, no se percibe una diferencia significativa, sin embargo, nos quedamos con el ARIMA(2,1,2) ya que presenta una leve ventaja respecto a los demás.

2.4. Modelo generalizado autorregresivo de heterocedasticidad condicional - GARCH. Un modelo GARCH de parámetros p y q o GARCH(p, q), se puede representar como:

$$(2.3) \quad \begin{aligned} \epsilon_t &= \vartheta_t \sigma_t \\ \sigma_t^2 &= \alpha_0 + \sum_{i=1}^p \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^q \gamma_j \sigma_{t-j}^2 \end{aligned}$$

donde $\{\vartheta_t\}$ es una sucesión de variables aleatorias independientes e idénticamente distribuidas con media cero y varianza 1, usualmente se asume que ϑ_t se distribuye

normal o t -student estándar, $\alpha_0 > 0$, y $\alpha_i \geq 0$, para $i \in \{1, 2, \dots, p\}$ garantizando que la varianza condicional σ_t^2 sea positiva [1, 19].

Estos modelos heterocedásticos condicionales se ocupan de la evolución en el tiempo de σ_t^2 , esto equivale a añadir una ecuación más que describe el comportamiento de la serie temporal [19].

Para construir el modelo GARCH nuevamente utilizaremos el paquete estadístico R. En primer lugar, es necesario especificar una ecuación para la media. Ya que tenemos un modelo ARIMA(2,1,2) que nos proporciona el mejor ajuste de la serie, este mismo modelo nos servirá para describir la media del proceso.

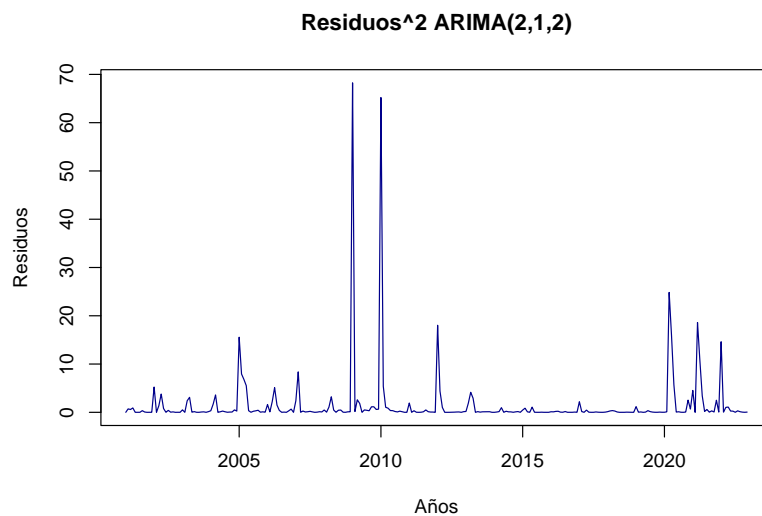


FIGURA 5. Volatilidad de la serie IMAE.

Como segundo paso, se debe comprobar si la varianza es heterocedástica y depende de los residuos al cuadrado rezagados, esto es llamado efecto ARCH [19]. Para ello, analizamos la ACF y PACF de los residuos al cuadrado que obtenemos del ARIMA, en el análisis realizado, los respectivos correlogramas no presentan dependencia lineal entre residuos, más aún, la prueba del multiplicador de Lagrange para identificar dicho efecto nos arroja un resultado no significativo, indicando nuevamente que nuestra serie no presenta heterocedasticidad condicional.

Bajo este análisis, los modelos GARCH no serían apropiados para modelar nuestra serie, ya que no cumple con todos los supuestos requeridos, por lo que proceder con las estimaciones mediante un modelo GARCH podrían tener poca validez.

2.5. Red neuronal multicapa. Los problemas de predicción de series de tiempo se consideran de aprendizaje supervisado, es decir, el modelo trabaja con datos de entrada-salida (etiquetados) para el entrenamiento, donde el conjunto de entrada son las variables autorregresivas de la misma serie temporal.

El conjunto de datos empleado para el problema se puede representar en forma matricial $\mathbf{D} = \{\mathbf{X}, \mathbf{Y}\}$, con:

$$(2.4) \quad \begin{aligned} \mathbf{X} &= [X^1, X^2, \dots, X^m], \\ \mathbf{Y} &= [Y^1, Y^2, \dots, Y^n] \end{aligned}$$

donde, \mathbf{X} es la matriz o conjunto de entradas, \mathbf{Y} es la matriz o conjunto de salidas, $X^i = \{x_k^i\}_{k=1, \dots, N}$ y $Y^j = \{y_k^j\}_{k=1, \dots, N}$, son los vectores de entradas y salidas, respectivamente, m y n son enteros que representan el número de variables de entrada y salida [9]. En nuestro caso, $n = 1$, ya que estamos analizando una serie temporal univariante y solo necesitamos una única variable de salida.

Con el análisis de los datos hecho previamente en los modelos estadísticos, podemos proceder de forma más rápida con la implementación de la red neuronal en Python.

2.5.1. Procesamiento de los datos. Sea $X = \{x_t\}$ la serie temporal, para el procesamiento de los datos, aplicamos la técnica de normalización,

$$x_i \text{ procesado} = \frac{x_i - \min(X)}{\max(X) - \min(X)},$$

con la cual pasamos a tener la serie de tiempo en una escala entre 0 y 1, esto es necesario puesto que las redes neuronales son sensibles cuando la escala de los datos es muy grande, trabajando de forma más adecuada si se reduce a un intervalo generalmente $[-1, 1]$ o $[0, 1]$.

También, es necesario construir la matriz de entradas \mathbf{X} , como se mencionó anteriormente, se trata de las mismas variables autorregresivas de la serie. El total de observaciones de nuestros datos es de 275, se debe definir un tamaño de ventana deslizante, es decir, el número de valores que tendrá cada columna de la matriz. Podemos relacionar este tamaño de ventana con los rezagos que se consideraban en los modelos ARIMA. Para este caso, definimos la ventana con un valor de 12, lo que equivale a tomar el primer rezago de la serie, quedando de la siguiente forma:

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_{12} \\ x_2 & x_3 & x_4 & \cdots & x_{13} \\ x_3 & x_4 & x_5 & \cdots & x_{14} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{263} & x_{264} & x_{265} & \cdots & x_{275} \end{bmatrix}$$

y la matriz de salidas:

$$\mathbf{Y} = [y_{12} \quad y_{13} \quad y_{14} \quad \dots \quad y_{275}]$$

2.5.2. Construcción y entrenamiento del modelo. Para la construcción del modelo, consideramos una capa de entrada de dimensión 12 (tamaño de la ventana), dos capas densas ocultas con 70 y 30 neuronas, respectivamente y ambas con función de activación tanh, por último, la capa de salida con una única neurona y función de activación lineal:

```
input_data = keras.Input(shape = (n_lag))
model_capas = layers.Dense(70, activation = "tanh")(input_data)
model_capas = layers.Dense(30, activation = "tanh")(model_capas)
model_capas = layers.Dense(1, activation = "linear")(model_capas)
modelo = keras.Model(input_data, model_capas)
```

Luego, para el entrenamiento se asignaron 150 épocas (epochs) con lotes de 30 valores, utilizando el optimizador Adam con tasa de aprendizaje de 0.001, sin barajar los datos, ya que al ser una serie temporal, se quiere captar el comportamiento que causa la dependencia en el tiempo.

2.6. Resultados. Para ambos modelos implementados se tomaron las observaciones desde el año 2001 al 2022 como conjunto de entrenamiento, las observaciones del último año de los datos, es decir, de enero 2023 a noviembre 2023 se tomaron como conjunto de validación (o prueba), comparando los valores observados de ese periodo con los predichos por el ARIMA(2,1,2) y la red neuronal multicapa.

En la tabla 2, se muestra la comparativa de los resultados del conjunto de prueba, mostrando los intervalos con 95 % de confianza y los errores de predicción.

Período	Observado	ARIMA(2,1,2)				Red Neuronal			
		Predicción	LI 95 %	LS 95 %	Error	Predicción	LI 95 %	LS 95 %	Error
2023-01-01	2.4	4.1457	1.6986	6.5928	1.7457	4.1753	1.9358	4.1682	1.7753
2023-02-01	3.1	3.9163	0.2363	7.5964	0.8163	2.5302	1.9259	4.1934	0.5698
2023-03-01	2.7	3.7155	-0.9773	8.4084	1.0155	2.9394	1.9229	4.1572	0.2394
2023-04-01	2.3	3.5454	-2.0009	9.0916	1.2454	2.6128	1.9018	4.1898	0.3128
2023-05-01	2.7	3.4064	-2.8522	9.6650	0.7064	2.2067	1.9428	4.1253	0.4933
2023-06-01	3.0	3.2981	-3.5433	10.1395	0.2981	2.5680	1.9439	4.2349	0.4320
2023-07-01	3.3	3.2189	-4.0874	10.5251	0.0811	2.7750	1.9513	4.1399	0.5250
2023-08-01	3.5	3.1664	-4.5001	10.8328	0.3336	3.1921	1.9480	4.1509	0.3079
2023-09-01	3.7	3.1378	-4.7985	11.0741	0.5622	3.2407	1.9160	4.1440	0.4593
2023-10-01	3.7	3.1299	-5.0010	11.2609	0.5701	3.7733	1.9322	4.1495	0.0733
2023-11-01	3.7	3.1395	-5.1258	11.4047	0.5605	3.5393	1.9592	4.1975	0.1607

TABLA 2. Valores observados y predichos del año 2023.

Como métricas de evaluación, consideramos el error cuadrático medio (MSE) y error absoluto medio (MAE) tanto para el conjunto de entrenamiento como el de prueba, los resultados presentados en la tabla 3, nos muestran una ventaja del modelo de red neuronal respecto al ARIMA(2,1,2) en capacidad predictiva. Esto también se logra apreciar gráficamente en la figura 6, donde vemos que el ajuste de los últimos valores es más preciso en la red neuronal.

	MSE training	MSE test	MAE training	MAE test
ARIMA(2,1,2)	1.529304	0.723384	0.656502	0.721370
Red Neuronal	1.718344	0.424903	0.755874	0.486248

TABLA 3. Métricas de evaluación de los modelos.

Comparación del Ajuste de los Modelos



FIGURA 6. Comparación gráfica de los modelos.

Los resultados obtenidos de la comparación entre el modelo ARIMA y la red neuronal multicapa para la estimación del IMAE de Honduras muestran diferencias significativas en cuanto a su capacidad predictiva. Si bien ambos modelos logran capturar la tendencia general de la serie temporal, la red neuronal muestra una mayor precisión en la predicción de los valores observados, sugiriendo que la aplicación de técnicas de aprendizaje profundo puede ofrecer mejoras en la estimación de series temporales económicas como el IMAE.

3. CONCLUSIONES

A continuación, se resume el trabajo realizado y los hallazgos observados durante la experimentación:

1. Se hizo un breve recorrido sobre el estudio de series de tiempo, destacando su importancia ya que con frecuencia nos encontramos con problemas en distintas ramas de la ciencia donde el modelado y pronóstico de una serie de tiempo es de gran utilidad, en especial, en la economía donde son aplicados una gran variedad de modelos con distintos enfoques de análisis prospectivo.
2. Se consideraron tres enfoques para el análisis de la serie de tiempo del indicador macroeconómico IMAE, el enfoque clásico con los modelos ARIMA, un enfoque de modelado de volatilidad con los modelos GARCH y un enfoque más moderno con las redes neuronales que en la actualidad son ampliamente utilizadas para la estimación y predicción mostrando buen rendimiento en estas tareas.

3. Los datos del IMAE a pesar de presentar dispersión, no presenta una volatilidad lo suficientemente alta como para ser capturada por un modelo GARCH, al no cumplir este supuesto lo recomendable es explorar otras alternativas donde se pueda contemplar la volatilidad de estos datos.
4. Se llevaron a cabo distintas pruebas para la elección de un modelo ARIMA, comparando métricas como el AIC, MSE y MAE, resultando un modelo ARIMA(2,1,2) como la mejor opción. Dicho modelo proporciona un ajuste aceptable para la serie de tiempo IMAE, sin embargo, a la hora de predecir muestra algunas discrepancias en ciertos períodos.
5. La red neuronal multicapa mostró una capacidad predictiva prometedora, con métricas de evaluación MSE y MAE que indican un buen ajuste, especialmente en el conjunto de prueba. Cabe mencionar que la red neuronal no necesita como tal que la serie de tiempo sea estacionaria o tenga una tendencia lineal, por lo que puede sorprender aún más esa capacidad para capturar el comportamiento de la serie IMAE, siendo una opción viable para el análisis de series de tiempo en entornos económicos.
6. Se realizó una comparación entre el modelo ARIMA(2,1,2) y la red neuronal multicapa en términos de métricas de desempeño y capacidad predictiva, proporcionando también una comparación visual entre la serie original y los valores del ajuste de los modelos. Con estas métricas se puede observar las fortalezas y limitaciones de cada uno, demostrando que explorar distintos enfoques para la estimación de series de tiempo resulta de gran utilidad.

Como trabajo a futuro se plantea la exploración de un enfoque bayesiano con los modelos lineales dinámicos. Además, posibles áreas de mejora o refinamiento de cada enfoque tratado en este trabajo, como un análisis de modelos ARIMA con residuos de colas pesadas, con el fin de mejorar el ajuste, en especial la predicción de este modelo. En el machine learning, explorar métodos de optimización de la arquitectura de una red neuronal y la implementación de una red neuronal LSTM con el fin de mejorar el aprendizaje entorno al indicador macroeconómico IMAE.

REFERENCIAS

1. Abelardo Monsalve, Pedro Harmath, *Introducción al análisis de series de tiempo con aplicaciones a la econometría y finanzas*, Escuela Venezolana de Matemáticas, Mérida, Venezuela, 2015.
2. A. Prat, D. Peña, M. Recober, *La teoría de series temporales, una evaluación crítica de los desarrollos mas recientes*, Question, Vol. 5 (1981), no. 3, 126-148.
3. Banco central de Honduras, *Metodología del cálculo del índice mensual de actividad económica, base 2000*.
4. Banco central de Honduras, *Índice mensual de actividad económica, enero 2023*, Departamento de estadísticas macroeconómicas, 2023.
5. Banco central de Honduras, *Serie Índice Mensual de Actividad Económica Base 2000=100*. Disponible en: <https://www.bch.hn/estadisticas-y-publicaciones-economicas/sector-real/indice-mensual-de-actividad-economica> (Revisado en: 25 Enero 2024).
6. Daniel Peña, *Análisis de series temporales*, Alianza Editorial S. A., Madrid, 2005.
7. Diego Broz y Valentina Viego, *Predicción de precios de productos de pinus spp. con modelos arima*, Madera y Bosques, vol.20 (2014), no. 1, 37-46.
8. George Box, Gwilym Jenkins, Gregory Reinsel and Greta Ljung, *Time series analysis: forecasting and control*, 5th edition, John Wiley & Sons, Inc., Hoboken, New Jersey, 2016.
9. Guillermo Nieto, *Predicción de series temporales mediante técnicas de aprendizaje automático, automatización del proceso*, Universidad de Sevilla, España, 2023.

10. Juan Abril, *Análisis de la evolución de las técnicas de series de tiempo: un enfoque unificado*, Instituto interamericano de estadística. Estadística **63** (2011), no. 181, 5-56.
11. J. Betancourt, J. García, R. Cepero y G. Rodríguez, *Predicción de estadía hospitalaria mediante modelo de serie de tiempo, caso hospital universitario*, Salud en Tabasco, Vol. **19** (2013), no. 1, 15-18.
12. John Kelleher, *Deep Learning*, The MIT press, 2019.
13. Jonathan Mairon y A. Quintero, *Predicción y categorización de series de tiempo usando una red neuronal multifuncional*, Fundación universitaria los libertadores, Bogota, 2022.
14. J. Otero y M. Sevilla, *Análisis de series temporales de observaciones gps en la estación permanente de Lanzarote*, Universidad complutense de Madrid, 2002.
15. Loren Trigo y Sabatino Costanzo, *Redes neuronales en la predicción de las fluctuaciones de la economía a partir del movimiento de los mercados de capitales*, El trimestre económico, vol. **74** (2007), no. 294, 415-440.
16. Marta Casas y Edilberto Cepeda. *Modelos arch, garch y egarch, aplicaciones a series financieras*, Cuadernos de Economía XXVII, no. **48** (2008), 287-320.
17. Ricardo Fernández, *Series temporales avanzadas: aplicación de redes neuronales para el pronóstico de series de tiempo*, Universidad de Granada, 2021.
18. Robert Engle, *Garch 101, the use of arch/garch models in applied econometrics*, Journal of economic perspectives, vol. **15** (2001), no. 4, 157-168.
19. Ruey Tsay, *Analysis of financial time series*, John Wiley & Sons, Inc, 3rd ed, 2010.

DEPARTAMENTO DE MATEMÁTICAS, UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS
 Dirección de correo electrónico: disa.vramos@gmail.com

FEEDFORWARD NEURAL NETWORKS, UNA EXPLICACIÓN GENERAL DE SU FUNCIONAMIENTO Y SU PAPEL EN MACHINE LEARNING

JOCSÁN ARIEL HERNÁNDEZ BARAHONA

RESUMEN. En la última década y sobre todo en los últimos cinco años, palabras como “Machine Learning”, “Deep Learning” e inteligencia artificial han estado mucho en auge. En este artículo daremos una reseña histórica de la creación de uno de los principales algoritmos que están en el núcleo de estos términos, las “Feedforward Neural Networks” (FNN). Detallamos los pormenores del algoritmo, en particular sus funciones de activación.

ABSTRACT. Over the last decade, and specially in the last five years, words like Machine Learning, Deep Learning and artificial intelligence have been upswinging. In this paper we will give a historic overview to the creation of one of the main algorithms that are core to all these terms, the Feedforward Neural Networks (FNN). We will give details about such algorithm and specially the activation functions.

1. INTRODUCCIÓN

El “machine learning” tiene su nacimiento quizá con el concepto de perceptón, también llamados neuronas de McCulloch–Pitts [13], un perceptón es un algoritmo de clasificación binaria, básicamente es un algoritmo capaz de distinguir, en su forma más básica entre dos estados [6]. Desde 1967 se utilizan algoritmos estocásticos para, mediante el uso de repetidos experimentos, un algoritmo pueda ser “entrenado” [8], con esto nos referimos a que el algoritmo será capaz de distinguir entre dos o más estados, lo que llamamos un algoritmo clasificatorio [6]. A partir del desarrollo de las computadoras y de los procesadores que cada vez son más veloces, el machine learning se ha dividido en dos tipos de redes neuronales, las “Feedforward” y las “Recurrent Neural Networks” [9]. Abordaremos en la sección de antecedentes la historia del desarrollo de las redes neuronales.

Uno de los principales problemas actuales con los sistemas capaces de tomar decisiones es saber cuáles de sus componentes pueden ser cambiadas o mejoradas para que la toma de decisiones sea más eficaz, es decir no sólo más veloz, sino también mejor, a este problema se le conoce como el problema fundamental de la asignación de crédito [6]. Si bien este es el problema general de machine learning, en este trabajo nos enfocaremos en el funcionamiento de las redes neuronales y su fundamento matemático, particularmente en las funciones de activación. Una red neuronal no es más que un conjunto de procesadores simples en conexión, a los que

Fecha: 26 de enero del 2024 y en revisión desde 14 de marzo, 2024.
Palabras y frases clave. Machine Learning, Deep Learning, Big Data.

individualmente se les conoce como neuronas [8].

Algunas neuronas pueden influenciar el ambiente, su funcionamiento básico se puede asemejar al de un grafo (dirigido o no) con pesos, ya sea activando neuronas cercanas o cambiando el peso de sus caminos [7]. La asignación de créditos o a lo que comúnmente se le llama “Learning” (aprendizaje) [8], se trata de asignar valores a los pesos tales que la red neuronal presente el comportamiento deseado, así sea actuar como un jugador en un videojuego o conducir un automóvil [15]. Dependiendo de la importancia de lo que se esté aprendiendo las redes neuronales pueden llegar a ser relativamente complejas, en las que existan transformaciones no lineales de varias etapas y pueden no estar definidas en la red neuronal primera, sino que se empiezan a transformar en cada etapa del computo [8]. El propósito del “deep learning” es asignar un valor adecuado a cada peso a lo largo de todos estos procesos [6]. En la sección XX desarrollamos un algoritmo en Python que implementa una “Feedforward Neural Network” (FNN de aquí en adelante) sencilla.

El machine learning es un tema en auge con muchísimas aplicaciones en muchas líneas de investigación [9], por supuesto ciencia y educación; pero con machine learning se pueden diseñar algoritmos de infraestructura y desarrollo territorial, pobreza e inequidad, globalización y un largo etc.[15]

Este trabajo pretende ser una explicación general de la fundamentación matemática de los algoritmos de FNN en las aplicaciones de machine learning y así estos puedan ser utilizados en diferentes aplicaciones.

2. ANTECEDENTES

Los inicios del machine learning (ML de aquí en adelante) y por tanto de las FNN datan de la publicación del libro “Preceptón” de 1958, escrito por el psicólogo Frank Rosenblatt [8], que después vendría a ser conocido como el padre del Deep Learning [12], el libro proponía una estructura matemática que consistía en nodos, llamados preceptones, que formaban una capa de entrada, una capa oculta con pesos aleatorios, sin capacidad de aprender, es decir cuyos pesos eran constantes una vez electos y una capa de salida que tenía la capacidad de aprender, con esto queremos decir que se proponía un algoritmo que podía cambiar su salida según sus entradas [11]. Como la capa oculta tiene pesos aleatorios inmutables, este sería el primer ejemplo de Extreme Machine Learning (EML) y no constituye como tal Deep Learning (DL), que es el funcionamiento actual del ML [6].

Pasaron 7 años, hasta 1965, para que el concepto de preceptón fuese tomado por matemáticos, particularmente Rusia tomaría los conceptos americanos para intentar desarrollar una inteligencia artificial bastante primitiva [8]. Fueron Alexey Grigorevich Ivakhnenko y Valentin Lapa, quienes en una publicación sobre teoría de grupos como herramientas para el manejo de datos, sentarían la fundamentación matemática para algoritmos que intentan optimizar varias variables al mismo tiempo [15]. Sin embargo, los métodos propuestos por Ivakhnenko y Lapa aún no eran iterativos, como tales tampoco eran demasiado precisos, excepto para problemas muy simples [8].

Es bien sabido que los procesos estocásticos forman gran parte del ML y ahora son impensables sin los mismos [11], dos años después del estudio de Ivakhnenko y Lapa, en Japón, Shunichi Amari y su estudiante de posgrado, Saito crearon la primera FNN con dos capas y la capacidad de aprender [12]. Según Amari los experimentos computacionales demostraron que la FNN podía distinguir entre dos patrones diferentes correctamente [11]. La FNN consistía de cinco capas, dos de las cuales eran de aprendizaje [12].

En 1970, 12 años después de la presentación del preceptón y en Finlandia, Seppo Linnainmaa, informático y matemático publica el método moderno de propagación hacia atrás [8], una aplicación del aprendizaje supervisado, que a su vez está basado en regla de la cadena, curiosamente Rosenblatt fue el primero en utilizar el término de propagación hacia atrás para referirse al aprendizaje de sus preceptones [7], aunque desconocía sobre como sería la implementación matemática de la misma.

Así mismo, vendría otra pausa de 12 años para que, en 1982, Paul Werbos, un científico social y pionero de ML, publicase una aplicación de la propagación hacia atrás tal y como se usa el día de hoy [8]; sin embargo los procesadores y capacidad de computación en general era muy baja como para poder ser utilizada en cualquier aplicación, inclusive las súper computadoras eran incapaces de hacer los experimentos en algún tiempo útil [7], es por ello que en 1985 y en un esfuerzo dirigido por David E. Rummerhart, psicólogo matemático, se intentó reducir la complejidad algorítmica de la propagación hacia atrás, con cierto éxito [4].

Por otro lado, la imposibilidad de implementar FNN en los noventa en computadoras con una, en aquel entonces, alta capacidad de procesamiento orilló a los investigadores a ponerse creativos, toda red neuronal es de hecho isomorfa a un grafo y estos a su vez isomorfos a un espacio matricial [8], utilizando este hecho Vladimir Vapnik y su equipo, todos matemáticos de la universidad de Uzbek, en Rusia, aprovechando la existencia de algoritmos rápidos para encontrar la nulidad de una matriz crearon el concepto de máquinas de vectores de soporte, como un perfecto análogo de las FNN, capaces de encontrar patrones en varias dimensiones a partir del cálculo de nulidad de varias matrices [7].

Fue hasta 2003 cuando resurge el interés en las FNN, con el científico computacional canadiense Yoshua Bengio quien crea una FNN capaz de interpretar lenguaje humano [8]. La FNN tenía la capacidad de, probabilísticamente, entender una serie de palabras y acatar ciertas órdenes según las mismas, a Bengio se le conoce como el padrino de la inteligencia artificial por este trabajo [15], es interesante que coloquialmente se le conoce como inteligencia artificial a cualquier FNN que pueda comprender lenguaje humano con cierta certeza [9].

Finalmente, en 2017 un equipo de investigadores dirigidos por el doctor en ciencias de la computación de la India, Ashish Vaswani, publicó una nueva forma de ver las FNN con la arquitectura de transformación moderna [8], la cual básicamente consiste en una FNN que carece de recurrencia y por lo tanto se entrena más rápido, la arquitectura previene cualquier bucle infinito o excesivamente largo [13], el modelo también consiste en aprendizaje de palabras a partir de grandes bancos

de información de palabras en varios idiomas, para el caso del artículo de Vaswani se basó en todos los artículos que existían en Wikipedia a la fecha [8].

3. MATEMÁTICAS INVOLUCRADAS EN LAS FNN

3.1. Funciones de activación. Una función de activación es una función matemática (cómo tal no es más que una 3-upla) que da un valor según su entrada y los pesos en los que se vea involucrada en un nodo [1].

Las dos funciones históricas que se han empleado en FNN como funciones de activación son las funciones sigmoideas (llamadas así por su forma en S) [5]:

$$(3.1) \quad y_{i+1}(v_i) = \tanh(v_i)$$

$$(3.2) \quad y_{i+1}(v_i) = \frac{1}{1 + e^{-v_i}}$$

El fin de utilizar funciones de tipo sigmoide, en particular las dos mencionadas arriba es, primero que nada por su rango de valores, que nos pueden devolver fácilmente de al menos 3 tipos; en el caso de la función 3.2 puede devolver 0, 1 o un valor intermedio ante cualquier entrada, así asignamos el valor 0 a evitar flujo de información (que se cierre el nodo y no envíe información al siguiente), asignamos el valor de 1 para promover el flujo total de información (que se envíe la salida a la siguiente capa de la red neuronal) y cualquier valor intermedio para “olvidar” [13], con olvidar nos referimos a que no se cierra el nodo; pero tampoco se envía la información, es decir se deja el nodo abierto a más información, sin importar lo anterior [10].

Los procedimientos de ML implican gradientes que se igualan a 0, esto para maximizar o minimizar los pesos y así obtener mejores salidas con las mismas entradas, el problema con la ecuación 3.2 es que su gradiente rápidamente tiende a 0, por lo que es difícil tener un proceso iterativo lo suficientemente robusto [1].

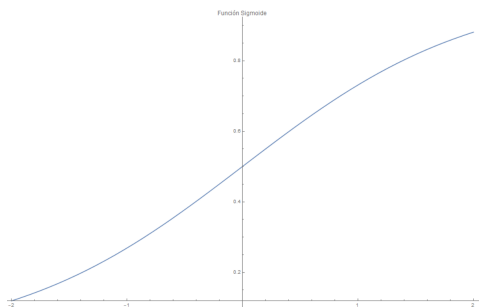


FIG. 1. La función sigmoide de la ecuación 3.2

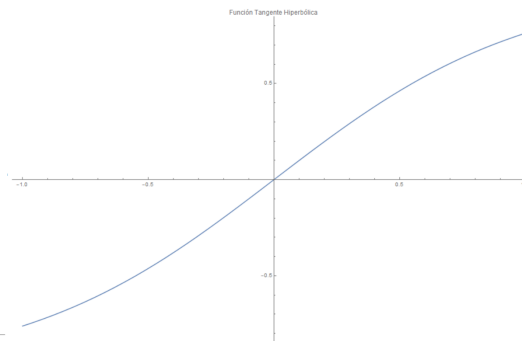


FIG. 2. La función tangente hiperbólica de la ecuación 3.1

En este sentido, una función asociada a un nodo neuronal debe ser acotada, diferenciable, monótona y de preferencia que sea fácilmente calculable, ya sea mediante métodos computacionales rápidos o directamente en su cálculo. Una función que cumple con todas estas características es la función de tangente hiperbólica, agregado al hecho de que existen muchos paquetes computacionales que calculan las derivadas y valores de la tangente hiperbólica con algoritmos de convergencia rápida [14].

Ya que la derivada de tangente hiperbólica no es más que:

$$(3.3) \quad \tanh'(z) = 1 - \tanh^2(z)$$

El cálculo de su primera y segunda derivada son de relativo bajo costo computacional. Agregado al hecho de que ahora tenemos un rango mayor de valores a los que podemos asignar cierto comportamiento, si tomamos su rango de -1 a 1, siendo -1 una decisión adicional a la presentada por la función sigmoide y tomar decisión de cerrar ciertos nodos adyacentes si obtenemos un valor intermedio positivo o negativo [10].

En recientes avances de ML y sobretodo gracias a que cada día se tienen computadoras con una capacidad de procesamiento mucho mayor [8], se ha desarrollado una nueva alternativa a las planteadas en las ecuaciones de arriba, se trata de la función de rectificación lineal, ReLU de aquí en adelante (por sus siglas en inglés “Rectified Lineal Unit”) [14], dicha función está definida como:

$$(3.4) \quad f(z) = \frac{x + |x|}{2}$$

Como se notará una de las principales desventajas de la función ReLU es que

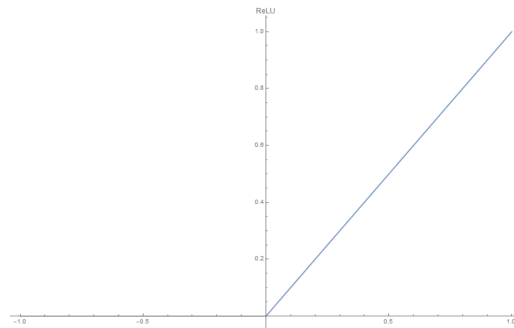


FIG. 3. La función ReLU de la ecuación 3.4

perdimos la capacidad de tomar múltiples decisiones, esto se puede solucionar teniendo más capas más sencillas [1]. Otra desventaja importante es que la segunda derivada de la función ReLU es 0 en todos los reales, haciendo difícil implementar el método del gradiente [13]. A cambio obtenemos la ventaja de que la función no crece tan rápido con valores grandes y ha encontrado una alta tasa de convergencia para aplicaciones de detección de imágenes [13].

3.2. Implementación de una función de activación. Presentaremos el código y explicación del mismo para la función de activación sigmoide, lo suficientemente sencilla como para explicar el concepto, recordemos que una FNN no es más que un algoritmo al que se le introduce cierta información y esperamos que haga algo [8], en este caso introduciremos un valor cualquiera y entrenaremos a la función sigmoide, para que si recibe un cero regrese un uno.

```

1  #Importamos modulos
2  import math
3  #Definiremos la función sigmoide
4  def sigmoide(x):
5      return 1/(1+math.exp(-x))
6
7  #definiremos una rutina para enseñarle a la función de activación que
   ↪ debe regresar un 1
8  #Usaremos una variable de ayuda para que la rutina termine
9  bandera=True
10 #usaremos un contador para ver cuántas iteraciones se necesitaron
11 contador=0
12 #Empezaremos con un valor de cero, ya que es lo que queremos lograr y
   ↪ aumentaremos o incrementaremos x, según si se acerca a uno o no
13 x_i=0
14 #Sólo haremos 50 iteraciones por segunda variable
15 #Debido a que la función sigmoide tiene puntos de equilibrio entre 0 y
   ↪ 1, necesitamos ir aumentando su valor de uno en uno cada 50
   ↪ iteraciones
16 n=0
17 while(bandera):
18     if(sigmoide(x_i)!=1):
19         x_i=sigmoide(x_i)+n
20         contador+=1
21     else:
22         bandera=False
23     if(contador==49):
24         n=n+1
25         contador=0
26         print("Enseñanza "+str(n))
27         print(sigmoide(x_i), "\t", contador)
28
29 print("El valor adecuado del peso es "+str(x_i))
30

```

En la línea 4 del código tenemos definida la función sigmoide de la ecuación [3.2], utilizamos una bandera en la línea 9 para saber cuando detener la enseñanza o aprendizaje general y un contador para iniciar otro ciclo de enseñanza. Nuestro valor inicial es 0 y recordemos que nuestra meta es modificar la función para que “sepa” que debe convertir 0’s en 1’s. Si la función es diferente de 1 asignamos al valor de x el valor de la función sigmoide más el número de enseñanza, que empezamos en 0, hacemos esto porque sabemos que la función sigmoide es monótonamente creciente al rededor de cero (ver figura 1), si logramos el cometido de obtener un valor de 1, la rutina de enseñanzas terminará. Si el ciclo de enseñanzas llega a 49

iteraciones aumentamos el contador. La función sigmoide tiene un comportamiento particular entre 0 y 1, la ecuación $x = \sigma(x)$ tiene varias soluciones en ese intervalo, donde $\sigma(x)$ es la función sigmoide y esto llevaría inevitablemente a un punto estable que impediría acercarnos al número 1 [1].

Agregamos además una tabla con la información de las rutinas (la primera y la última). Podemos ver que después de 37 ciclos de enseñanza de 50 iteraciones, obtenemos lo esperado, si se introduce un 0, obtenemos un 1.

Rutina 1		Rutina 36	
Valor	Iteración	Valor	Iteración
0.6224593312018546	1	0.9999999999999998	1
0.6507776782147005	2	0.9999999999999998	2
0.6571856892772512	3	0.9999999999999998	3
0.6586279095209511	4	0.9999999999999998	4
0.6589521000392728	5	0.9999999999999998	5
0.6590249529909402	6	0.9999999999999998	6
0.6590413236655800	7	0.9999999999999998	7
0.6590450022424830	8	0.9999999999999999	8
0.6590458288354719	9	1	9
0.6590460145745493	10	1	10

TABLA 1. Resultado de los ciclos de enseñanza, todos con un valor inicial de 0

3.3. Método del descenso del gradiente. Una analogía que siempre se utiliza para introducir este tema [6] es la siguiente: imagine que está perdido en un bosque, sin mapa ni brújula y que lo único que sabe es que debe llegar a la parte más baja del bosque para salir del mismo, ¿cómo se logra? Simplemente se toman pasos en la dirección que apunte más hacia abajo, la más inclinada hacia abajo. En esta analogía el bosque serían todos los parámetros posibles de un problema y la parte más baja los valores que nos ayudan a minimizar o maximizar una función [2], sabemos que estos dos problemas son de hecho equivalentes y el dual el uno del otro [3].

3.3.1. Función de costo. Llamamos función de costo a una función que mide el rendimiento de un modelo matemático para ciertos datos [8], la función de costo es el corazón de ML ya que nos dirá que tan bien se ajustan nuestros datos al modelo propuesto [7]. La función de costo cuantifica el error entre los valores predichos de un modelo matemático y los valores esperados de los datos, su co-dominio es el espacio de todos los parámetros y su dominio son los números reales [15].

Para ilustrar la idea veamos un ejemplo sencillo en el que aplicaremos un modelo de tipo lineal a datos, normalmente es buena idea graficar los datos, en caso de poderse [2] y luego elegir un modelo que se adecúe a los mismos, también existe la posibilidad de hacer un ML a la elección del modelo; pero no es el propósito de este artículo. Empezamos con un modelo al que llamaremos hipótesis [15] y supongamos

que los datos se comportan de una forma aproximadamente lineal, nuestra hipótesis sería entonces:

$$(3.5) \quad h_x(\theta_0, \theta_1) = \theta_1 x + \theta_0$$

Evidentemente los parámetros de nuestro modelo son los números reales θ_1, θ_0 . Para FNN normalmente utilizamos el método de los mínimos cuadrados para la función de costo del modelo [15], teniendo así:

$$(3.6) \quad J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m \left(h_{\theta}(x^{(i)}) - y^{(i)} \right)^2,$$

donde J es la función de costo, m es el total de datos, en este caso θ es un vector de parámetros, x es el valor a evaluar en la función hipótesis y y es el valor de los datos, aquí i representa el i -ésimo dato para la función de costo, la meta es minimizar la función de costo $J(\theta_0, \theta_1)$ mediante el cambio de sus parámetros [9].

3.3.2. *El método en FNN.* Cómo dijimos en la sección de antecedentes, FNN retoma su fuerza hasta que se consiguen computadoras lo suficientemente potentes como para utilizar métodos iterativos, en este caso se varían los parámetros en la dirección contraria al gradiente, para obtener el mayor decrecimiento posible de la función de costo [15].

El fin del método es encontrar un conjunto de parámetros que minimice la función de costo y así obtener un modelo lo más preciso posible para un conjunto de datos [2]. El algoritmo funciona calculando el gradiente de la función de costo y luego varía los parámetros en la dirección contraria al gradiente [6], en FNN normalmente el modelo tiene que ver con una función de activación [15], aunque el método del descenso del gradiente se puede aplicar de forma general a cualquier forma de ML [7].

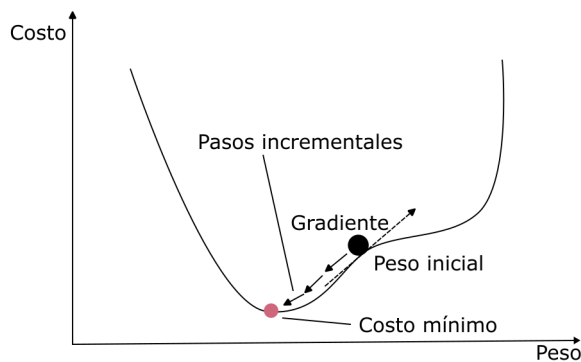


FIG. 4. Ilustración del algoritmo del descenso del gradiente

El algoritmo de descenso del gradiente funciona con dos pasos iterativos [2]:

- (1) Se calcula el gradiente de la función de costo.
- (2) Se cambian los parámetros en dirección opuesta al gradiente.

3.3.3. Algoritmo de descenso del gradiente.

$$\theta_{j+1} = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

En el caso de nuestra hipótesis tenemos únicamente dos parámetros, por ello es que j tiene únicamente a 0 y 1 por valores permitidos [2]. A α se le conoce como un parámetro de aprendizaje y determina qué tan grande es el cambio en los parámetros en dirección opuesta al gradiente [6]. Cuando $\theta_{j+1} = \theta_j$ o al menos lo es dentro de los límites de un error de convergencia establecido, el algoritmo termina [15].

Este algoritmo fue, de hecho, propuesto por Agustin-Louis Cauchy en 1847 [2]. Sin embargo, como mencionamos en la sección de historia fue hasta los años 2000 cuando el método es utilizado para FNN y ML [7].

Existen muchas variaciones del método de descenso del gradiente, entre ellas el descenso de gradiente por partes, descenso del gradiente estocástico, descenso del mini-gradiente por partes, entre otros [2] cada uno con sus ventajas y limitaciones [1] es trabajo del que está haciendo el modelaje y la FNN como tal, el elegir un método adecuado al problema que se le presenta [15].

4. IMPLEMENTACIÓN DE UNA FNN SENCILLA

Ahora que tenemos toda la teoría haremos una implementación sencilla de la misma; primero una red neuronal sencilla mostrada en la siguiente figura realizada por el autor:

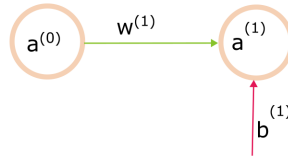


FIG. 5. Una FNN de dos neuronas

En la figura la primera capa de neuronas está determinada por el súper-índice 0, que como es común en computación el 0 indica la primera posición [3], w aquí indica la función de activación o peso y b indica un parámetro de ajuste conocido como parámetro de sesgo [15]. Este parámetro de sesgo sirve para evitar las soluciones de la forma $\sigma(x) = x$, donde $\sigma(x)$ es la función de activación [13].

Vamos a entrenar ahora a nuestra red neuronal para que sea una función de negación, es decir que si recibe un 0 regrese un 1 y si recibe un 1 regrese un 0, se puede ver esta implementación en el siguiente código de Python implementado por el autor:

```

1 #importando modulos
2 import numpy as np

```

```

3  #Declaramos nuestra función de activación como una tangente
   ↪ hiperbólica
4  activacion=np.tanh
5  #Definimos nuestros parámetros de sesgo y peso
6  w1=-10
7  b1=10
8  #Definimos el comportamiento de nuestra neurona
9  def a1(a0):
10     return activacion(w1*a0+b1)
11
12 #Verificamos el comportamiento
13 print(a1(0))
14 print(a1(1))

```

```

PS D:\Jocsan\Clases\Matemáticas\Seminario de investigación\FNN> python3.12 .\funcion_no.py
0.9999999958776927
0.0

```

FIG. 6. Resultado del código para la red neuronal de la figura 7

Este ha sido un ejemplo muy simple del funcionamiento de una FNN, por supuesto que esta puede llegar a ser tan compleja como requiera una aplicación [15], en dado caso los pesos los representamos por una matriz y los parámetros de sesgo se convierten en un vector [7], finalmente en la siguiente sección entrenamos datos en una FNN y lo combinamos con el método del descenso del gradiente.

5. FNN PARA ENTRENAR DATOS

Utilizaremos el método de descenso del gradiente y una FNN de cuatro capas (dos ocultas, una de entrada y una de salida) para hacer un algoritmo capaz de replicar un dibujo, en este caso un corazón, para el caso de las gráficas y como no se trata de una gráfica con varios mínimos locales [15], usaremos la función sigmoide 3.2 para entrenar los datos.

Para capturar correctamente la complejidad de la curva utilizaremos dos capas ocultas de 6 y 7 neuronas, respectivamente, la FNN tomará un dato (el valor en x de la función) y dará dos datos (el par ordenado de coordenadas en el plano xy), como se muestra en la figura 9 implementada por el autor.

Sabemos que en las FNN cada capa recibe información de la anterior según nuestra función de activación [15], así nuestras neuronas siguen el siguiente patrón:

$$a^{(n)} = \sigma(z^{(n)}),$$

donde:

$$z^{(n)} = W^{(n)}a^{(n-1)} + b^{(n)}.$$

A continuación se puede ver la implementación del código y sus resultados

```

1  #Implementación de una FNN que aprende a dibujar un corazón
2  #Importamos modulos
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import matplotlib.colors as mcolors

```

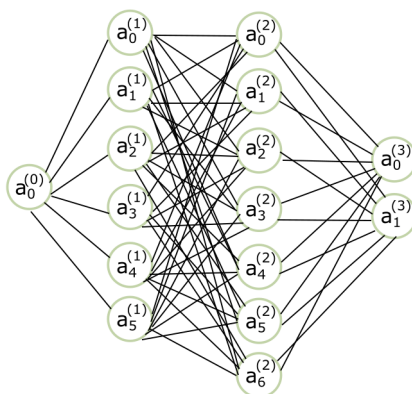



FIG. 7. FNN de dos capas ocultas

```

6  #Definimos nuestra función de activación y su derivada
7  sigma = lambda z : 1 / (1 + np.exp(-z))
8  d_sigma = lambda z : np.cosh(z/2)**(-2) / 4
9  #Empezamos definiendo una función que crea una FNN inicial y la
   ↪ reinicia en caso de necesitarlo
10 #Los pesos se crean de forma aleatoria
11 #Estos serán cambiados a medida la FNN vaya aprendiendo, así que
   ↪ simplemente escogemos
12 #algunos valores para los pesos y b es el parámetro de sesgo para cada
   ↪ capa de la red
13 def reiniciar_fnn (n1 = 6, n2 = 7, random=np.random) :
14     global W1, W2, W3, b1, b2, b3
15     W1 = random.randn(n1, 1) / 2
16     W2 = random.randn(n2, n1) / 2
17     W3 = random.randn(2, n2) / 2
18     b1 = random.randn(n1, 1) / 2
19     b2 = random.randn(n2, 1) / 2
20     b3 = random.randn(2, 1) / 2
21 #Definimos la FNN de la siguiente forma:
22 def funcion_fnn(a0) :
23     z1 = W1 @ a0 + b1
24     a1 = sigma(z1)
25     z2 = W2 @ a1 + b2
26     a2 = sigma(z2)
27     z3 = W3 @ a2 + b3
28     a3 = sigma(z3)
29     return a0, z1, a1, z2, a2, z3, a3
30 #Definimos nuestra función de costo como la suma de los errores al
   ↪ cuadrado del modelo con respecto a los datos
31 def cost(x, y) :
32     return np.linalg.norm(funcion_fnn(x)[-1] - y)**2 / x.size
33

```

```

34 #Implementamos el método del gradiente para cada parámetro de la FNN,
    ↪ lo hacemos por aparte para más claridad
35 # Jacobiano para la tercera capa
36 def J_W3 (x, y) :
37     # Primero asignamos todos los valores que regresa la función
38     a0, z1, a1, z2, a2, z3, a3 = funcion_fnn(x)
39     # Calculamos la derivada parcial del costo respecto a a3
40     J = 2 * (a3 - y)
41     #Usamos regla de la cadena para calcular el Jacobiano
42     J = J * d_sigma(z3)
43     # Calculamos el gradiente para cada punto de los datos del eje x
44     J = J @ a2.T / x.size
45     return J
46
47 #Calculamos el Jacobiano para el sesgo de la tercera capa
48 def J_b3 (x, y) :
49     a0, z1, a1, z2, a2, z3, a3 = funcion_fnn(x)
50     #Hacemos lo mismo que en el primer caso; pero para el sesgo
51     J = 2 * (a3 - y)
52     J = J * d_sigma(z3)
53     # Aplicamos el jacobiano para cada capa
54     J = np.sum(J, axis=1, keepdims=True) / x.size
55     return J
56
57 #Un proceso idéntico sigue la segunda capa
58 def J_W2 (x, y) :
59     a0, z1, a1, z2, a2, z3, a3 = funcion_fnn(x)
60     J = 2 * (a3 - y)
61     J = J * d_sigma(z3)
62     J = (J.T @ W3).T
63     J = J * d_sigma(z2)
64     J = J @ a1.T / x.size
65     return J
66
67 def J_b2 (x, y) :
68     a0, z1, a1, z2, a2, z3, a3 = funcion_fnn(x)
69     J = 2 * (a3 - y)
70     J = J * d_sigma(z3)
71     J = (J.T @ W3).T
72     J = J * d_sigma(z2)
73     J = np.sum(J, axis=1, keepdims=True) / x.size
74     return J
75
76 #Y la capa primera sigue un proceso idéntico
77 def J_W1 (x, y) :
78     a0, z1, a1, z2, a2, z3, a3 = funcion_fnn(x)
79     J = 2*(a3-y)
80     J = J * d_sigma(z3)
81     J = (J.T @ W3).T
82     J = J * d_sigma(z2)
83     J = (J.T @ W2).T
84     J = J*d_sigma(z1)

```

```

85     J = J @ a0.T / x.size
86     return J
87
88
89 def J_b1 (x, y) :
90     a0, z1, a1, z2, a2, z3, a3 = funcion_fnn(x)
91     J =2*(a3-y)
92     J = J * d_sigma(z3)
93     J = (J.T @ W3).T
94     J = J * d_sigma(z2)
95     J = (J.T @ W2).T
96     J =J*d_sigma(z1)
97     J = np.sum(J, axis=1, keepdims=True) / x.size
98     return J
99
100 #Debemos definir nuestra función que hará las veces de un corazón
101 def plot_training (x, y, iterations=10000, aggression=3.5, noise=1) :
102     global W1, W2, W3, b1, b2, b3
103     fig,ax = plt.subplots(figsize=(8, 8), dpi= 80)
104     ax.set_xlim([0,1])
105     ax.set_ylim([0,1])
106     ax.set_aspect(1)
107
108     xx = np.arange(0,1.01,0.01)
109     yy = np.arange(0,1.01,0.01)
110     X, Y = np.meshgrid(xx, yy)
111     Z = ((X-0.5)**2 + (Y-1)**2)**(1/2) / (1.25)**(1/2)
112     im = ax.imshow(Z, vmin=0, vmax=1, extent=[0, 1, 1, 0],
113     ↪ cmap=blueMap)
114
115     ax.plot(y[0],y[1], lw=1.5, color=green)
116     fig.show()
117
118     while iterations>=0 :
119         j_W1 = J_W1(x, y) * (1 + np.random.randn() * noise)
120         j_W2 = J_W2(x, y) * (1 + np.random.randn() * noise)
121         j_W3 = J_W3(x, y) * (1 + np.random.randn() * noise)
122         j_b1 = J_b1(x, y) * (1 + np.random.randn() * noise)
123         j_b2 = J_b2(x, y) * (1 + np.random.randn() * noise)
124         j_b3 = J_b3(x, y) * (1 + np.random.randn() * noise)
125
126         W1 = W1 - j_W1 * aggression
127         W2 = W2 - j_W2 * aggression
128         W3 = W3 - j_W3 * aggression
129         b1 = b1 - j_b1 * aggression
130         b2 = b2 - j_b2 * aggression
131         b3 = b3 - j_b3 * aggression
132
133     if (iterations%100==0) :
134         nf = funcion_fnn(x)[-1]
135         ax.plot(nf[0],nf[1], lw=2, color=magentaTrans);

```

```

136     iterations -= 1
137
138     nf = funcion_fnn(x)[-1]
139     ax.plot(nf[0],nf[1], lw=2.5, color=orange)
140
141
142 def training_data (N = 100) :
143     x = np.arange(0,1,1/N)
144     y = np.array([
145         16*np.sin(2*np.pi*x)**3,
146         13*np.cos(2*np.pi*x) - 5*np.cos(2*2*np.pi*x) -
147         ↪ 2*np.cos(3*2*np.pi*x)- np.cos(4*2*np.pi*x)
148     ]) / 20
149     y = (y+1)/2
150     x = np.reshape(x, (1, N))
151     #y = np.reshape(y, (2, N))
152     return x, y
153
154 def make_colormap(seq):
155     seq = [(None,) * 3, 0.0] + list(seq) + [1.0, (None,) * 3]
156     cdict = {'red': [], 'green': [], 'blue': []}
157     for i, item in enumerate(seq):
158         if isinstance(item, float):
159             r1, g1, b1 = seq[i - 1]
160             r2, g2, b2 = seq[i + 1]
161             cdict['red'].append([item, r1, r2])
162             cdict['green'].append([item, g1, g2])
163             cdict['blue'].append([item, b1, b2])
164     return mcolors.LinearSegmentedColormap('CustomMap', cdict)
165
166 magenta = (0xfc/255, 0x75/255, 0xdb/255)
167 magentaTrans = (0xfc/255, 0x75/255, 0xdb/255, 0.1)
168 orange = (218/255, 171/255, 115/255)
169 green = (175/255, 219/255, 133/255)
170 white = (240/255, 245/255, 250/255)
171 blue1 = (70/255, 101/255, 137/255)
172 blue2 = (122/255, 174/255, 215/255)
173
174 blueMap = make_colormap([blue2, blue1])
175
176
177 #Obtenemos los datos a entrenar propios de la figura de un corazón
178 x, y = training_data()
179 reiniciar_fnn()
180 #Finalmente entrenamos nuestra red neuronal unas 10 mil veces
181 #Y haremos unos 10 entrenamientos
182 for i in range(10):
183     plot_training(x, y, iterations=10000, aggression=3, noise=0)
184 input()

```

Si bien el código está comentado es importante destacar algunos puntos, primero que nada tenemos una función que se encarga de reiniciar la FNN al asignar pesos aleatorios y parámetros aleatorios a los sesgos; pero siempre se intenta que los iniciales sean números relativamente pequeños, esto se ve de la línea 13 a la 20. De la línea 22 a la 29 se define el comportamiento de la FNN según los patrones descritos justo arriba del código, en esta misma sección. Tenemos de las líneas 31 a la 32 nuestra función de costo. De la línea 36 a las 100 definimos el método de descenso del gradiente a partir de los jacobianos para cada capa de la función de costo, a partir de sus parámetros que es justo esto lo que buscamos minimizar. De la línea 103 a la 176 entrenamos los datos según el número de iteraciones, la agresión y el ruido, estos son parámetros extra que se pueden definir según se quiera en la red neuronal, la agresión está relacionada con el parámetro de aprendizaje definido en la sección anterior y el ruido tiene que ver con qué tan bien se puede ajustar los datos a un modelo, si el ruido es muy alto es posible que se requiera eliminar muchos datos.

Por último, presentamos los resultados del primer entrenamiento y del décimo, ya que hemos ejecutado 10, de 10 mil iteraciones:

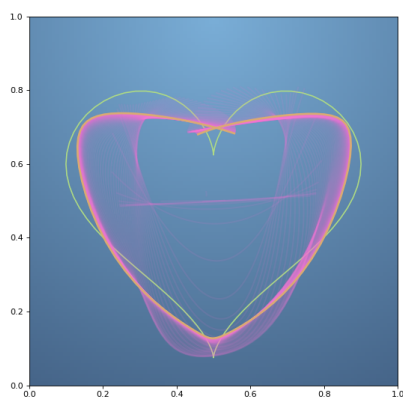


FIG. 8. Primer entrenamiento

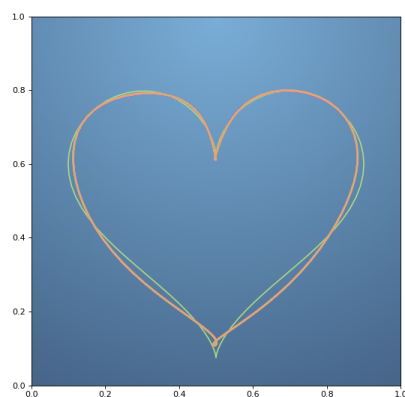


FIG. 9. Décimo entrenamiento

Como podemos ver en la figura 9, se obtiene un resultado mejor después de 10 entrenamientos, si aumentamos el número de iteraciones por entrenamientos o el número de entrenamientos, se obtendrán mejores resultados.

6. CONCLUSIONES

Hemos dado un breve viaje por los principales hechos históricos y avances del ML, viendo que este es más antiguo que cuando se popularizó en los principios de la década del 2010 (ver sección de antecedentes). Hemos explicado el funcionamiento matemático y los principios que rigen las FNN, si bien existen otras formas de crear redes neuronales, las FNN son más sencillas de entender como un primer acercamiento a redes neuronales, como hemos mencionado existen otras, que tienen

mayores requerimientos computacionales. Hemos especificado el algoritmo general de funcionamiento de las FNN combinado con el algoritmo del descenso del gradiente.

Finalmente en este artículo presentamos un algoritmo que sirve para replicar un modelo basado en datos, en una red neuronal de 4 capas, produciendo buenos resultados. Esto se logró mediante la implementación de varios parámetros en cada una de las capas, sin embargo se podrían obtener mejores resultados haciendo la red neuronal de más capas ocultas, aumentando así la cantidad de parámetros. Como trabajo a futuro se implementaría una FNN con más capas y más neuronas intermedias para mejorar la réplica de los datos a entrenar. De la misma manera sólo presentamos 10 entrenamientos; pero con una computadora más potente se podrían implementar hasta 1000 entrenamientos.

Como trabajo a futuro, también, nos gustaría implementar otras formas de redes neuronales más potentes en servidores en la nube, así como implementar FNN en el modelado de problemas de índole nacional, en particular un modelado del tráfico vehicular en Tegucigalpa, como ser lugares y horas de mayor tráfico para identificar las causas y proponer soluciones al mismo.

REFERENCIAS

1. A. Krizhevsky, I. Sutskever, et al. *ImageNet classification with deep convolutional neural networks*. Communications of the ACM. **60** (6) (2017) 84–90. doi:10.1145/3065386.
2. C. Lemaréchal, *Cauchy and the Gradient Method* Doc Math **Extra** 251–254 (2012) preprint.
3. C. R. Harris, K. J. Millman; Stéfan J, et al, *Array programming with NumPy*. Nature. **585** (7825) 357–362. doi:10.1038/S41586-020-2649-2.
4. David E. Rumelhart et al., *Learning Internal Representations by Error Propagation*, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, MIT Press, 1987.
5. H. Guang-Bin, Z. Qin-Yu, S. Chee-Kheong *Extreme learning machine: theory and applications*. Neurocomputing. **70** (1) (2006): 489–501. doi:10.1016.
6. J. Kelley, *Gradient theory of optimal flight paths*. ARS Journal. 30 (10) (1960), 947–954. doi:10.2514/8.5282.
7. J. Schmidhuber, *Annotated History of Modern AI and Deep Learning* The SwissAI Lab IDSIA (Grenoble) **02-22** (2022), 1-88, arXiv:2212.11279 (preprint).
8. J. Schmidhuber, *Deep Learning in Neural Networks: An Overview* The Swiss AI Lab IDSIA (Grenoble) **03-14** (2014), 1-88, arXiv:1404.7828 (preprint).
9. K. Hinkelmann, *Neural Networks* University of Applied Sciences Northwestern Switzerland **2** (2018), 1-63 (preprint)
10. S. Amari, *A Theory of Adaptive Pattern Classifiers* IEEE Transactions on Electronic Computers, **EC-16**, no. **3**, (1967) 299-307, doi: 10.1109.
11. T. Charles, *Who is the Father of Deep Learning?*. International Conference on Computational Science and Computational Intelligence (CSCI). IEEE. **2019** (2019) 343–348. doi:10.1109/CSCI49370.2019.00067.
12. W. Duch, N. Jankowsky, *Survey of Neural Transfer Functions* Neural Computing Surveys **2** (1999), 163-212. doi=10.1.1.43.6996
13. X. Glorot, A. Bordes, et al. *Deep sparse rectifier neural networks*. International Conference on Artificial Intelligence and Statistics. **1** (2011) 1-9. (preprint)
14. Y. Freund, R. Schapire, *Large margin classification using the perceptron algorithm* Machine Learning. **37** (3) (1999), 277–296. doi:10.1023.
15. Y. LeCun, Y. Bengio, G. Hinton, *Deep Learning* Nature. **521** (7553) 436–444. doi:10.1038/nature14539.

FNN Y SU PAPEL EN MACHINE LEARNING

ESCUELA DE MATEMÁTICAS, FACULTAD DE CIENCIAS, UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS, TEGUCIGALPA MDC, HONDURAS 11101

Dirección actual: Escuela de Matemáticas, Facultad de Ciencias, Universidad Nacional Autónoma de Honduras, Tegucigalpa MDC, Honduras 11101

Dirección de correo electrónico: jocsan.hernandez@unah.hn

INVARIANTES POLINOMIALES DE NUDOS

ALAN SAMUEL HERNÁNDEZ FLORES

RESUMEN. La teoría de nudos es una rama relativamente joven de la topología, que comenzó a ser estudiada formalmente en los siglos XVIII y XIX. Una de las primeras motivaciones para investigar los nudos fue la propuesta de una teoría que sugería que los átomos no eran otra cosa más que nudos en una sustancia llamada éter. Aunque esta teoría fue finalmente descartada, el interés en el estudio de los nudos persistió, y en la actualidad se conocen aplicaciones en campos como la química y la biología. A lo largo de este trabajo, nos enfocaremos en explorar conceptos fundamentales. Iniciaremos con la definición de nudo, examinaremos los movimientos de Reidemeister, abordaremos la equivalencia de nudos y exploraremos invariantes polinomiales de nudos, tales como el polinomio de Alexander y el polinomio de Jones.

ABSTRACT The theory of knots is a relatively young branch of topology, which began to be formally studied in the 18th and 19th centuries. One of the earliest motivations for investigating knots was the proposal of a theory suggesting that atoms were nothing more than knots in a substance called ether. Although this theory was ultimately discarded, interest in the study of knots persisted, and nowadays applications are known in fields such as chemistry and biology. Throughout this work, we will focus on exploring fundamental concepts. We start with the definition of a knot, examine Reidemeister moves, address knot equivalence, and explore polynomial knot invariants, such as the Alexander polynomial and the Jones polynomial.

1. INTRODUCCIÓN

Los nudos han estado presente a través de los años, según descubrimientos arqueológicos que revelan que la práctica de los nudos se remonta a la prehistoria. Además de sus aplicaciones cotidianas para unir objetos, los nudos han atraído el interés humano debido a su estética. Estos se pueden apreciar en varias obras artísticas de antiguas civilizaciones.

Pese a que los nudos han sido utilizados desde hace mucho tiempo, el estudio matemático de los nudos es algo reciente. La teoría de nudos fue motivado por la química. En la década de 1880, se creía que una sustancia llamada éter impregnaba todo el espacio. En un intento de explicar los diferentes tipos de materia, Lord Kelvin hipotetizó que los átomos eran nudos en el éter. Diferentes nudos corresponderían a diferentes elementos. Esto convenció al físico escocés Peter Guthrie Tait de que si pudiera listar todos los nudos posibles, estaría creando una tabla de los elementos. Pasó muchos años tabulando nudos, pero lamentablemente la teoría de Kelvin fue descartada. Un modelo más preciso de la estructura atómica apareció

Fecha: Abril 16, 2024.

Palabras y frases clave. Topología, nudo, invariante de nudo, polinomio de Alexander, polinomio de Jones.

a fines del siglo XIX y los químicos perdieron interés en los nudos, pero mientras tanto, los matemáticos se habían intrigado con los nudos [1].

Uno de los problemas centrales en la teoría de nudos radica en determinar cuándo dos nudos son equivalentes y cuándo son claramente diferentes. Una forma de clasificar nudos es a través de invariantes. En este estudio, nos sumergiremos en una introducción a la teoría de nudos, centrándonos específicamente en la investigación de dos invariantes recientes de nudos, tales como el polinomio de Alexander y el polinomio de Jones. El polinomio de Alexander fue introducido por James Waddell Alexander en 1928, constituyendo el primer invariante polinómico de nudos y el polinomio de Jones es un polinomio de nudo descubierto por Vaughan Jones en 1984.

A pesar de ser una teoría abstracta han surgido aplicaciones relacionadas con el ADN de los virus, bacterias y mitocondrias de las células humanas, en las cuales el ADN suele ser una molécula circular que se enrolla. Este enrollamiento impide el acceso a la información a menos que se actúe sobre la doble hélice para separar las hélices. Si las dos hebras del ADN no estuvieran entrelazadas, sería fácil separarlas simplemente empujando cada una en una dirección diferente. Pero no es así, y lo que ocurre en las bacterias es que una de las hebras se enrolla sobre la otra, de manera que forman un círculo. Pero para poder ser útiles, necesitan desenrollarse para permitir el acceso a la información y la transcripción de genes. La teoría de nudos se ha demostrado muy útil para entender la acción de las enzimas llamadas Topoisomerasas en todo ese proceso de enrollamiento y desenrollamiento [17].

2. ANTECEDENTES

Una de las primeras incursiones hacia el estudio de lo que ahora llamamos teoría de nudos fue la realizada por Carl Friedrich Gauss, quien en 1833 desarrolló la integral que calcula el número de enlaces de dos curvas. Así, dadas dos curvas diferenciables y parametrizadas $\gamma_1, \gamma_2 : S^1 \rightarrow \mathbb{R}^3$ la integral de enlace de Gauss viene dada por

$$Lk(\gamma_1, \gamma_2) = \frac{1}{4\pi} \oint_{\gamma_1} \oint_{\gamma_2} \frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3} \cdot (d\mathbf{r}_1 \times d\mathbf{r}_2)$$

donde \mathbf{r}_1 y \mathbf{r}_2 son los vectores de posición a lo largo de las curvas, y $d\mathbf{r}_1$ y $d\mathbf{r}_2$ son los elementos de línea infinitesimales a lo largo de las curvas.

El trabajo de Gauss inspiró a otros matemáticos a adentrarse en el estudio de la teoría de nudos. Johann Benedict Listing, quien estudió con Gauss en Göttingen durante la década de 1830, desarrolló un interés por los nudos durante sus estudios de topología. Listing publicó “Vorstudien zur Topologie”, un artículo dedicado en parte al estudio de los nudos. El enfoque particular de Listing se centraba en la quiralidad de los nudos, es decir, la equivalencia entre un nudo y su imagen especular. Un resultado significativo presentado en su artículo fue la afirmación de que el nudo trébol no es anfiquiral, es decir, no es equivalente a su imagen reflejada. Sin embargo, esta afirmación no fue demostrada hasta 1914 por Max Dehn, quien utilizó información que era recientemente publicada sobre el grupo de nudos. Posteriormente, Listing afirmó que el nudo en forma de ocho y su imagen especular son equivalentes, o sea, son anfiquirales [13].

Posteriormente, motivado por la hipótesis de Lord Kelvin, en la década de 1880 Peter Tait realizó los primeros artículos sobre la clasificación de nudos. Aunque

Tait es a menudo reconocido por sus primeras tabulaciones de nudos, fue el matemático Thomas Kirkman quien realizó la primera contribución importante a la tarea de clasificar nudos. Thomas Kirkman estaba interesado casi exclusivamente en la tabulación de nudos alternantes e hizo una tabla de diagramas para nudos alternantes con hasta once cruces. Durante su búsqueda para clasificar diferentes nudos, Kirkman se dio cuenta de que tendría que reducir sus diagramas de nudos para minimizar el número de duplicados, por lo que ideó un método que involucraba una operación similar al segundo movimiento de Reidemeister para simplificar sus diagramas resultando así en una tabla de nudos mas precisa [13].

Tait se asoció con Charles Newton Little, profesor de la Universidad Estatal de Nebraska, para continuar el trabajo de Kirkman en la tabulación de nudos. Experimentaron con los diferentes métodos de notación, incluida la notación de Listing, finalmente se decidieron por una versión ligeramente diferente de la notación de Listing que alteraron para eliminar la ambigüedad. Tait y Little finalmente descubrieron algunos nudos repetidos en las tablas de Kirkman, y tras algunas modificaciones en los diagramas publicaron la primera tabla oficial de nudos alternos con hasta diez cruces [5]. Las investigaciones de Tait llevaron al desarrollo de lo que ahora se conoce como las conjeturas de Tait.

En la década de 1920 el matemático Emil Artin desarrollo la teoría de trenzas, la cual dio lugar a que un matemático interesado en la teoría de nudos notara una posible conexión entre nudos y trenzas. James Waddell Alexander descubrió el primer invariante polinomial de nudos en 1928, lo que le permitió distinguir muchos nudos no isotópicos entre si [1]. Basó su polinomio en el concepto de grupo de trenzas de Artin, así como en información recientemente publicada contenida en el libro escrito sobre teoría de nudos de Kurt Reidemeister [7].

Kurk Reidemeister se interesó por la teoría de nudos en la década de 1920, y su trabajo sobre el tema se centró en gran medida en diagramas planos de nudos. Inicialmente, Reidemeister lucho por crear un nuevo método de clasificación de nudos. Evaluó varias técnicas, incluido un esfuerzo por representar nudos con ecuaciones. Cuando ningún enfoque analítico ni combinatorio proporcionó información adecuada para crear un diagrama del nudo o manipular el nudo, Reidemeister recurrió al método de clasificar por diagramas [18]. Reidemeister, utilizando diagramas de nudos similares a los publicados por Tait, Little y Kirkman demostró que dos nudos K y K' con diagramas D y D' son equivalentes si y solo si sus diagramas están relacionados por una secuencia finita de diagramas intermedios, de manera que cada uno difiere de su predecesor por uno de los tres movimientos de Reidemeister [16].

El teorema de Reidemeister requería una demostración complicada y proporcionó a los teóricos de los nudos un resultado esencial que ayudaría a establecer las invariantes de los nudos. Reidemeister comenzó con la suposición de que dos nudos, K y K' , son equivalentes y que cada uno tiene un diagrama equivalente, D y D' respectivamente. Sabiendo que se puede obtener K' de K mediante una deformación de \mathbb{R}^3 sobre sí mismo (lo que se conoce como una isotopía ambiental) se deduce que se puede obtener D' de D utilizando un número finito de operaciones. Reidemeister demostró que es posible dividir cada diagrama en partes de modo que cada parte contenga un solo arco o un solo cruce. A partir de este punto, analizó todas las operaciones posibles que podrían realizarse en estos fragmentos del diagrama

y finalmente demostró que las únicas posibilidades eran los tres movimientos de Reidemeister [16].

Reidemeister demostró que tres operaciones eran suficientes para representar la isotopía ambiental [18]. Resulta interesante notar que los movimientos empleados por Reidemeister fueron previamente definidos por Maxwell varios años antes. El aspecto significativo del trabajo de Reidemeister no radicó en demostrar que era posible transformar un nudo en otro equivalente utilizando los movimientos de Reidemeister, sino en demostrar que estos tres movimientos eran los únicos necesarios para ilustrar la equivalencia entre dos nudos. Esto se convirtió en un factor crucial para el desarrollo de invariantes de nudos [5].

En la década de 1980 el matemático Vaughan Jones desarrolló una nueva invariante polinomial. Jones demostró la invariancia de su polinomio bajo isotopía ambiental; el polinomio de Jones ahora podría usarse para distinguir dos nudos no equivalentes entre sí. Este fue el único polinomio invariante descubierto desde el polinomio de Alexander, y resultaría muy poderoso [5].

El polinomio de Jones demostró ser un invariante más sensible que el de Alexander. Después de su descubrimiento, el polinomio de Jones se calculó para nudos de hasta trece cruces. Todos estos nudos estaban asociados con polinomios únicos, con la excepción de dos nudos, cada uno con once cruces. Un examen meticuloso de estos dos nudos reveló que eran equivalentes y se corrigió la tabla de nudos [18].

El hallazgo del polinomio de Jones resultó ser crucial para la teoría de nudos, representando el primer invariante polinómico capaz de discernir la lateralidad de un nudo. Específicamente, tiene la capacidad de distinguir entre el trébol derecho y el trébol izquierdo, una distinción imposible de lograr con el polinomio de Alexander. Además, el polinomio de Jones se utilizó en la primera demostración rigurosa de las conjeturas de Tait. Aunque el polinomio de Jones es un invariante sensible, no es completo, ya que existen nudos no isotópicos que comparten el mismo polinomio de Jones.

3. CONCEPTOS BÁSICOS DE LA TEORÍA DE NUDOS

3.1. Definiciones básicas. Podemos conceptualizar un nudo matemático de manera intuitiva imaginando tomar un trozo de cuerda, realizar un nudo en él y luego unir los extremos de la cuerda para crear un bucle con el nudo. Ver Figura 3.1.



FIGURA 3.1. Fuente: Elaboración propia.

Para realizar un estudio matemático formal de los nudos debemos pasar de esta idea intuitiva a una definición formal. Antes de llegar a esta definición, es pertinente presentar algunos conceptos preliminares.

Definición 3.1. Sean X, Y espacios topológicos y $f : X \rightarrow Y$; entonces, f es un homeomorfismo si se cumple que:

- f es biyectiva
- f es continua
- La inversa de f es continua.

Definición 3.2. Una M -variedad es un espacio de Hausdorff X con una base numerable tal que cada punto $x \in X$ tiene un entorno que es homeomorfo a un subconjunto abierto de \mathbb{R}^M .

Una 1-variedad se denomina una curva y una 2-variedad se denomina una superficie. Un ejemplo de una 1-variedad que será de mucha importancia en nuestro estudio es el círculo unitario \mathbb{S}^1 .

Definición 3.3. Sean X, Y espacios topológicos y $f : X \rightarrow Y$. Diremos que f es un encaje o embebimiento si $f : X \rightarrow f(X)$ es un homeomorfismo.

Un ejemplo de encaje es la función $f : [0, 2\pi) \rightarrow \mathbb{R}^2$ definida como $f(\theta) = (\cos(\theta), \sen(\theta))$

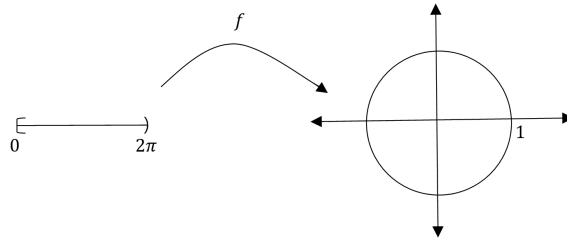


FIGURA 3.2. Encaje de $[0, 2\pi)$ en \mathbb{R}^2 . Fuente: Elaboración propia.

Teorema 3.1. Si X es una M -variedad compacta, entonces X se puede embeber en \mathbb{R}^N para algún entero positivo N [11].

Si tomamos nuestra 1-variedad compacta \mathbb{S}^1 por el teorema anterior tenemos que se puede embeber en \mathbb{R}^N para algún entero positivo N . Esto nos permite dar la siguiente definición de nudo.

Definición 3.4. Un nudo es un encaje $K : \mathbb{S}^1 \rightarrow \mathbb{R}^3$.

El nudo mas simple es la circunferencia estándar \mathbb{S}^1 vista en \mathbb{R}^3 , es decir, el conjunto de puntos $\{(x, y, 0) \in \mathbb{R}^3 \mid x^2 + y^2 = 1\}$ en $\mathbb{R}^2 \subset \mathbb{R}^3$. Este nudo es llamado nudo trivial, es simplemente el círculo desanudado. Ver Figura 3.3.

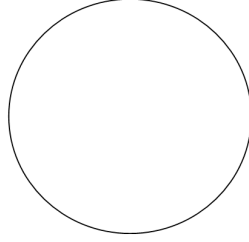


FIGURA 3.3. Nudo trivial. Fuente: Elaboración propia.

Ahora nos podemos preguntar ¿porqué en la definición de nudo se tomó $N = 3$? Es decir, ¿Por qué no definimos un nudo K como un subespacio de \mathbb{R}^n que sea homeomorfo a \mathbb{S}^1 ? Tenemos que n debe ser mayor o igual que dos, ya que toda aplicación continua de la circunferencia en la recta real manda por lo menos un par de puntos diametralmente opuestos al mismo punto [9] y por lo tanto $f : \mathbb{S}^1 \rightarrow \mathbb{R}$ no definiría un homeomorfismo. Si $N = 2$ tendríamos que los encajes serían curvas cerradas simples que topológicamente son iguales. Si $N > 3$ podríamos desanudar todos los nudos, esto debido a que al agregar más dimensiones se pueden hacer movimientos que en \mathbb{R}^3 provoquen rupturas o intersecciones de la curva y que en las dimensiones altas son permitidas y provocan que se desenrede el nudo. Y es así como la teoría de nudos adquiere una mayor riqueza cuando los nudos están inmersos en el espacio tridimensional \mathbb{R}^3 .

Uno de los problemas fundamentales en la teoría de nudos consiste en determinar cuándo dos nudos son equivalentes. Observe que, la cuestión no se reduce a determinar si dos nudos K_1 y K_2 son homeomorfos, ya que todos son homeomorfos a \mathbb{S}^1 y, por lo tanto, también lo son entre sí. Existen dos definiciones diferentes de equivalencia de nudos:

Definición 3.5. Dos nudos K_1 y K_2 son equivalentes si existe un homeomorfismo $h : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ tal que $h(K_1) = K_2$.

La otra definición alternativa de equivalencia es más estricta. Se basa en una construcción más elaborada que modela el proceso de transformar un nudo en otro, esta se describe mediante la definición de isotopía de ambiente.

Definición 3.6. Una isotopía de ambiente entre dos aplicaciones continuas $f, g : X \rightarrow Y$ es otra aplicación continua $H : X \times [0, 1] \rightarrow Y$ que cumpla $H(x, 0) = f(x)$, $H(x, 1) = g(x) \forall x \in \mathbb{R}^3$ y tal que si $t \in [0, 1]$ entonces $H(x, t) = h_t(x)$ es un homeomorfismo.

Con lo cual, una deformación isotópica de un espacio topológico X es una familia de homeomorfismos h_t , $0 \leq t \leq 1$, de X en sí mismo, tal que $h_0(x) = f(x)$ para todo x en X , y la función H definida por $H(x, t) = h_t(x)$ es simultáneamente continua en t y x .

Se dice que los nudos K_1 y K_2 pertenecen al mismo tipo de isotopía si existe una deformación isotópica $\{h_t\}$ de \mathbb{R}^3 tal que $h_1 \circ K_1 = K_2$. La letra t se elige

intencionalmente para sugerir el tiempo. Así, para un punto fijo x en \mathbb{R}^3 , el punto $h_t(x)$ traza, por así decirlo, la trayectoria de la molécula originalmente en x durante el movimiento de la cuerda desde su posición inicial en K_1 hasta K_2 .

Note que, si los nudos K_1 y K_2 pertenecen al mismo tipo de isotopía, son equivalentes. Sin embargo, con la definición de equivalencia que tenemos hasta ahora la afirmación recíproca es falsa. La siguiente discusión sobre la orientación sirve para ilustrar la diferencia entre las dos definiciones.

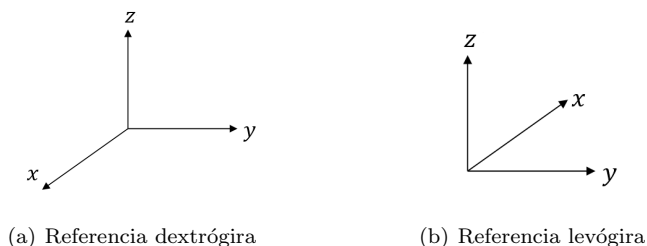


FIGURA 3.4. Fuente: Elaboración propia.

Todo homeomorfismo h de \mathbb{R}^3 en sí mismo preserva la orientación o invierte la orientación. Intuitivamente h invierte la orientación si transforma una referencia dextrógira de \mathbb{R}^3 en una referencia levógira (Figura 3.4) y preserva la orientación si la deja igual. La composición de homeomorfismos sigue las reglas:

h_1	h_2	$h_1 \circ h_2$
preserva	preserva	preserva
invierte	preserva	invierte
preserva	invierte	invierte
invierte	invierte	preserva

Es claro que, la aplicación identidad preserva la orientación. Por otro lado, la reflexión $(x, y, z) \mapsto (x, y, -z)$ la invierte. Si h es una transformación lineal, preserva o invierte la orientación dependiendo si su determinante es positivo o negativo. De manera similar, si tanto h como su inversa son diferenciables en cada punto de \mathbb{R}^3 , entonces h conserva o invierte la orientación según su jacobiano sea siempre positivo o siempre negativo.

Consideremos una deformación isotópica $\{h_t\}$ de \mathbb{R}^3 . El hecho de que la identidad preserve la orientación, combinado con la continuidad de $H(x, t) = h_t(x)$, sugiere que h_t preserva la orientación para cada t en el intervalo $0 \leq t \leq 1$. Esto es cierto. Como resultado, tenemos que una condición necesaria para que dos nudos sean del mismo tipo de isotopía es que exista un homeomorfismo de \mathbb{R}^3 en sí mismo que preserve la orientación y que mapee un nudo en el otro [6]. Tomando esto en cuenta, damos nuestra segunda definición de equivalencia de nudos.

Definición 3.7. Se dice que dos nudos K_1 y K_2 son equivalentes si existe una isotopía de ambiente entre la identidad y un homeomorfismo h que transforme K_1 en K_2 , es decir, una isotopía $F : \mathbb{R}^3 \times [0, 1] \rightarrow \mathbb{R}^3$ de forma que $F(x, 0) = x$ y

$F(x, 1) = h(x) \forall x \in \mathbb{R}^3$ y $h(K_1) = K_2$. La equivalencia de nudos la denotaremos por $K_1 \simeq K_2$.

Esta definición es equivalente a decir que dos nudos K_1 y K_2 son equivalentes si existe un homeomorfismo $h : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ que preserve la orientación tal que $h(K_1) = K_2$.

Las definiciones 3.5 y 3.7 difieren entre sí y ofrecen dos clasificaciones distintas. Por ejemplo, según la definición 3.5, el trébol y su imagen especular (Figura 3.5) son considerados equivalentes, aunque no son nudos isotópicos. Ambas definiciones se utilizan según el contexto. Por ejemplo, en las tablas de nudos primos, es decir, nudos que no se pueden descomponer en una suma conexas, un nudo y su imagen especular se agrupan en la misma clase de nudos en la clasificación, lo que correspondería al uso de la primera definición. Por otro lado, en las invariantes polinómicas de nudos, se hace una distinción entre el trébol y su imagen especular. En el resto de este trabajo usaremos la definición 3.7 a menos que se diga lo contrario.

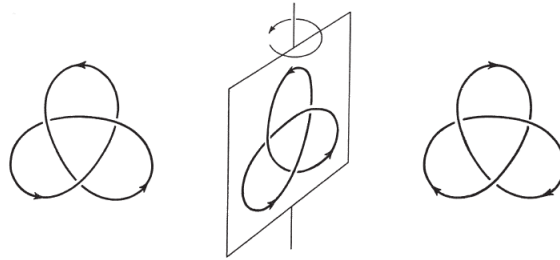
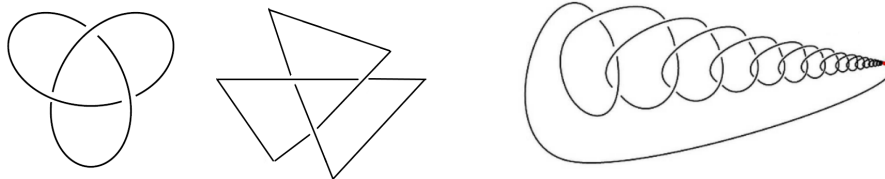


FIGURA 3.5. Trébol y su imagen especular. Tomado de [6].

3.2. Diagramas Regulares.

Definición 3.8. Un nudo poligonal es aquel formado por una unión finita de segmentos de recta llamados aristas, cuyos puntos extremos son los vértices del nudo. Un nudo es dócil si es equivalente a un nudo poligonal. Los nudos que no son dóciles se llaman salvajes.



(a) Nudo dócil. Fuente: Elaboración propia.

(b) Nudo salvaje. Tomado de [12].

FIGURA 3.6

Un nudo generalmente es representado mediante una proyección en un plano. Consideremos la proyección dada por:

$$p : \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

$$p(x, y, z) = (x, y, 0)$$

Si tomamos un nudo K diremos que $p(K) = \widehat{k}$ es la proyección de K . Si K tiene una orientación, su proyección también hereda la orientación. Notemos que \widehat{K} no es una curva cerrada simple, ya que posee varios puntos de intersección, estos puntos en los que suceden estas intersecciones los solemos llamar un punto de cruce; mas formalmente un punto q de \widehat{K} es llamado un punto de cruce si la imagen inversa $p^{-1}(q) \cap K$ contiene mas de un punto de K . El orden de $q \in \widehat{K}$ es la cardinalidad de $p^{-1}(q) \cap K$. Así, un punto doble es un punto de cruce de orden 2, un punto triple es uno de orden 3 y así sucesivamente [6].

Las proyecciones de nudos pueden volverse complicadas cuando se presentan puntos de cruce de órdenes superiores a 2, ya que determinar el número de cruces únicamente observando la proyección se vuelve más difícil. Por esta razón, buscamos proyecciones donde solo existan puntos de cruce de orden 2, a las cuales denominamos proyecciones regulares. Para un nudo poligonal, las proyecciones más simples son aquellas en las que los nudos se encuentran en posición regular.

Definición 3.9. Un nudo K está en posición regular si su proyección satisface lo siguiente:

- Los únicos puntos de cruce de \widehat{K} son puntos dobles.
- Ningún punto doble es la imagen de ningún vértice de K .

La segunda condición asegura que todo punto doble represente un punto de cruce genuino como en la Figura 3.7 (a); los puntos dobles como en Figura 3.7 (b) están prohibidos [6].

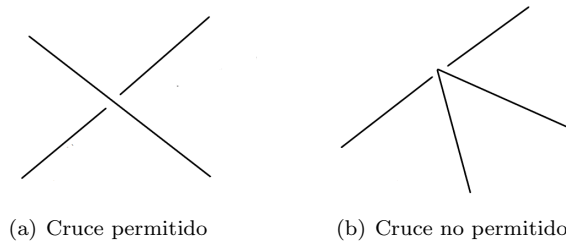


FIGURA 3.7. Tomado de [6].

Teorema 3.2. *Cualquier nudo poligonal K es equivalente, bajo una rotación arbitrariamente pequeña de \mathbb{R}^3 , a un nudo poligonal en posición regular [6].*

Por lo tanto, todo nudo poligonal puede ser representado por una proyección regular. De ahora en adelante trabajaremos exclusivamente con proyecciones regulares. Si

a estas proyecciones le dibujamos los cruces de forma que se vea que trozo del nudo pasa por debajo y cual pasa por encima tal y como se ve en la figura 3.8 le llamaremos diagramas regulares.

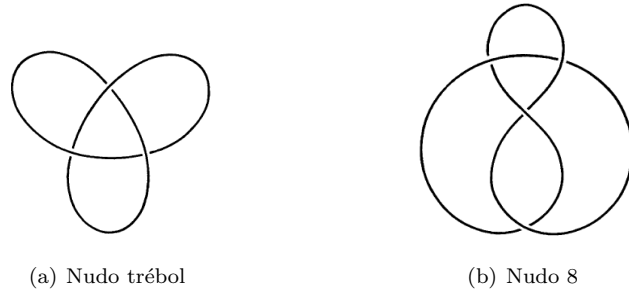


FIGURA 3.8. Ejemplos de diagramas regulares. Tomado de [6].

3.3. Movimientos de Reidemeister. Suele resultar bastante difícil proporcionar una isotopía de manera explícita y también demostrar su inexistencia, por lo tanto, optaremos por emplear otras estrategias. Como vimos en la sección anterior, muchas veces es útil proyectar los nudos en el plano y después estudiarlos mediante sus diagramas regulares. Para hacer esto, necesitamos preguntarnos cómo se transforma el diagrama regular de un nudo al llevarlo a través de una isotopía hacia otro nudo equivalente en una posición regular. Esta pregunta fué estudiada por Kurt Reidemeister en la década de 1920 y describió dichas reglas que ahora se conocen como movimientos de Reidemeister:

- (I) El primer movimiento nos permite agregar o quitar un giro en el nudo.
- (II) El segundo movimiento nos permite agregar o quitar 2 cruces.
- (III) El tercer movimiento nos permite deslizar un segmento del nudo de un lado de un cruce al otro lado del cruce.

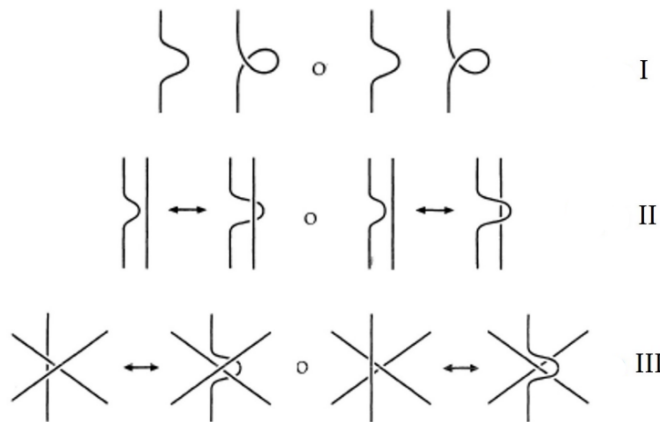


FIGURA 3.9. Movimientos de Reidemeister. Tomado de [3].

Cada movimiento cambia la proyección del nudo, pero no cambia el nudo representado por la proyección. Cada uno de estos movimientos es una isotopía ambiental. Reidemeister demostró que estos tres movimientos junto con equivalencias topológicas planas de los diagramas son suficientes para generar la isotopía espacial (en \mathbb{R}^3).

Definición 3.10. Dos diagramas de nudo son llamados equivalentes si se puede pasar de uno al otro mediante una sucesión finita de movimientos de Reidemeister.

Teorema 3.3. *Dos nudos son equivalentes si y solo si sus diagramas son equivalentes [15].*

En la Figura 3.10 se puede ver como del primer diagrama se llega al nudo trivial.

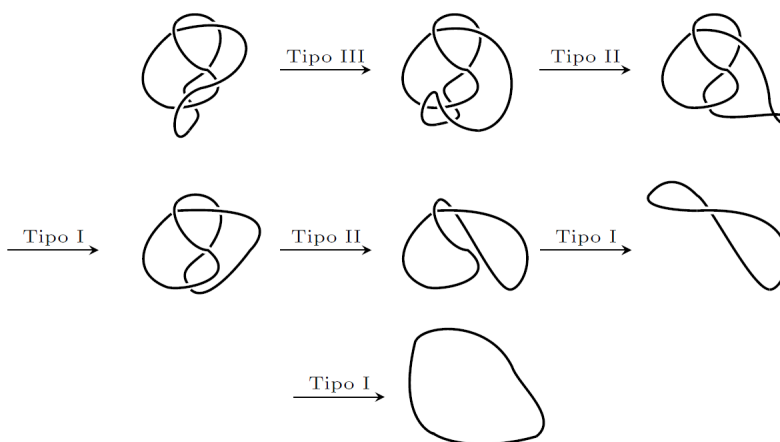


FIGURA 3.10. Verificando equivalencia de diagramas. Tomado de [12].

3.4. Enlaces.

Definición 3.11. Un enlace es una colección ordenada finita de nudos que no se intersectan entre si. Cada nudo K_i se dice que es una componente del enlace.

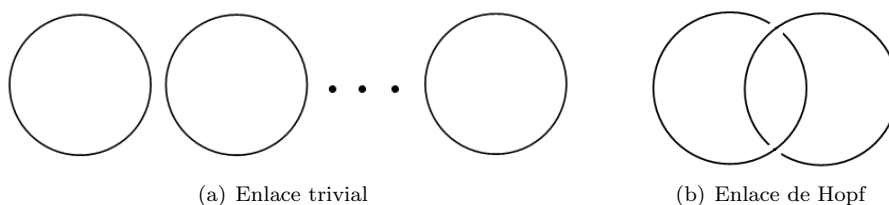


FIGURA 3.11. Ejemplos de enlace. Fuente: Elaboración propia.

Un nudo se considera un enlace de un solo componente. Si dos proyecciones representan el mismo enlace, debe haber una secuencia de movimientos de Reidemeister para ir de una proyección a la otra.

Definición 3.12. Dos enlaces $L = \{K_1, K_2, \dots, K_m\}$ y $L' = \{K'_1, K'_2, \dots, K'_n\}$ son equivalentes si se satisfacen las siguientes condiciones:

- 1) $m = n$, es decir, L y L' tienen el mismo número de componentes.
- 2) Existe un homeomorfismo de \mathbb{R}^3 en sí mismo que preserva la orientación que manda la colección $K_1 \cup K_2 \cup \dots \cup K_m$ en la colección $K'_1 \cup K'_2 \cup \dots \cup K'_n$.

3.5. Nudos anfiqueirales y nudos invertibles.

Definición 3.13. Un nudo K se dice que es anfiqueiral si existe un homeomorfismo $h : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ que invierte la orientación tal que $h(K) = K$.

Definición 3.14. La imagen especular de un nudo K es la imagen de K bajo la reflexión \mathcal{R} definida por $(x, y, z) \mapsto (x, y, -z)$.

Una formulación equivalente de la definición, que es más atractiva geoméricamente, se proporciona mediante el siguiente lema.

Lemma 3.4. *Un nudo K es anfiqueiral si y solo si existe un homeomorfismo que preserve la orientación de \mathbb{R}^3 en \mathbb{R}^3 que mapea K en su imagen especular [6].*

Demostración. Si K es anfiqueiral la composición $\mathcal{R} \circ h$ preserva la orientación y mapea a K en su imagen especular. Por el contrario si h' es un homeomorfismo que preserva la orientación de \mathbb{R}^3 sobre sí mismo que mapea K en su imagen especular, entonces la composición $\mathcal{R} \circ h'$ invierte la orientación y $(\mathcal{R} \circ h')(K) = K$.

Un ejemplo de un nudo anfiqueiral es el nudo 8. En la Figura 3.12 se muestra el proceso de obtener su imagen especular. Cuando un nudo no es anfiqueiral, se le denomina quiral. Como mencionamos en nuestra discusión sobre la equivalencia de nudos, el nudo trébol no es isotópico a su imagen especular, lo que significa que es un nudo quiral.

Del mismo modo en que cualquier homeomorfismo de \mathbb{R}^3 en sí mismo preserva o invierte la orientación, lo mismo ocurre con cualquier homeomorfismo de K en sí mismo. Al asignar una dirección al nudo K , la función f preserva o invierte la orientación si mantiene o invierte el orden de los puntos de K .

Definición 3.15. Un nudo K es invertible si existe un homeomorfismo de \mathbb{R}^3 en sí mismo que preserva la orientación, tal que la restricción $h|_K$ es un homeomorfismo que invierte la orientación de K en sí mismo.

El nudo trébol y el nudo 8 son nudos invertibles.

INVARIANTES POLINOMIALES DE NUDOS

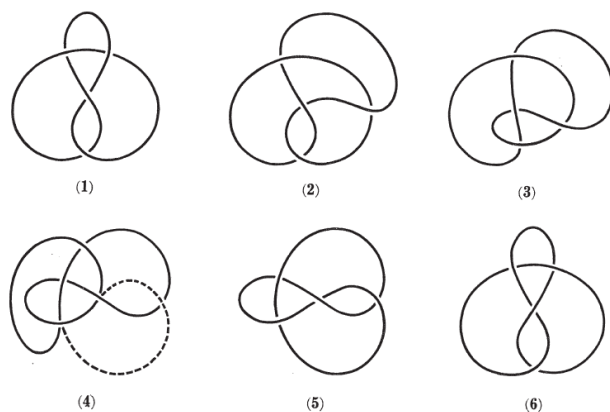


FIGURA 3.12. Equivalencia del nudo 8 y su imagen especular. Tomado de [6].

4. INVARIANTES POLINOMIALES

4.1. Polinomio de Alexander. James Waddell Alexander, un matemático estadounidense nacido en Nueva Jersey en 1888, realizó sus estudios en Matemáticas y Física en la Universidad de Princeton, donde completó su doctorado en 1915. Su contribución más destacada a la teoría de nudos fue el desarrollo de un invariante polinómico que se puede calcular a partir del diagrama de un nudo. Para construir este polinomio, seguiremos la ruta combinatoria que Alexander utilizó [2].

Para calcular el polinomio de Alexander comencemos con un diagrama orientado D de un nudo K . Supongamos que hay v puntos de cruce en el diagrama que denotaremos como c_1, c_2, \dots, c_v . Entonces por el teorema de Euler sobre poliedros, se sigue que los arcos del diagrama dividen el plano en $v + 2$ regiones, incluyendo la región fuera del nudo. Etiquetamos las regiones como r_0, r_1, \dots, r_{v+1} .

En cada punto de cruce, dos de las cuatro esquinas estarán marcadas con puntos para indicar que de las dos ramas a través del punto de cruce se debe considerar como la que pasa por debajo, o detrás, de la otra mientras sigue la orientación dada del nudo. Ver Figura 4.1.

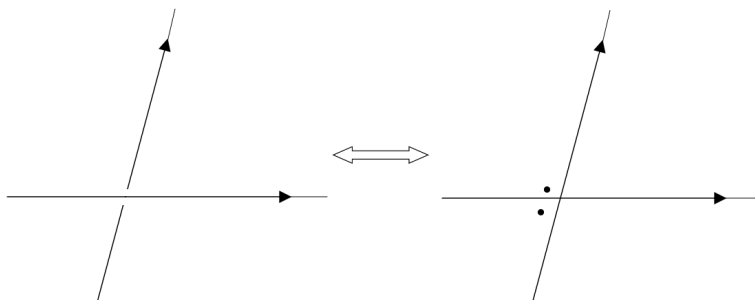


FIGURA 4.1. Representación de un cruce. Fuente: Elaboración propia.

Ahora, examinemos un punto de cruce arbitrario, denotado como c_i . Supongamos que las cuatro regiones que lo rodean son r_j , r_k , r_l y r_m , y que alrededor del punto de cruce avanzamos en sentido contrario a las agujas del reloj. Los puntos están ubicados en las regiones r_j y r_k (Figura 4.2). Con esto establecido, podemos definir la siguiente ecuación lineal:

$$c_i(r) = tr_j - tr_k + r_l - r_m = 0,$$

tomando una suma alternada de los símbolos que representan las cuatro regiones en su orden cíclico y multiplicando las regiones punteadas por t .

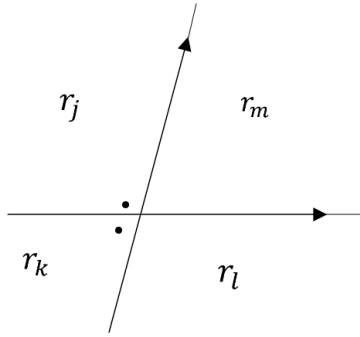


FIGURA 4.2. Fuente: Elaboración propia.

Definir una ecuación para cada cruce en el diagrama da lugar a un sistema de v ecuaciones en $v + 2$ variables. Estas ecuaciones pueden representarse en una matriz M de $v \times (v + 2)$, donde cada entrada es $\pm t$, ± 1 o 0 . En esta matriz, cada fila corresponde a un punto de cruce del diagrama y cada columna corresponde a una región. El siguiente paso consiste en seleccionar dos regiones vecinas r_p y r_q y eliminar sus columnas correspondientes v_p y v_q de la matriz. Pueden elegirse dos regiones vecinas arbitrarias, y se puede demostrar que la elección de estas regiones no afectará al invariante resultante [2] [10].

Al eliminar las columnas v_p y v_q , nos queda una matriz cuadrada de $v \times v$, $M_{p,q}$. La matriz $M_{p,q}$ se llama la matriz de Alexander del nudo k . Ahora denotemos por $\Delta_{p,q}(t)$ sea el determinante de esta matriz cuadrada, que será un polinomio en potencias de t con coeficientes enteros.

Teorema 4.1. *El polinomio $\Delta_{p,q}(t)$ obtenido como se describió anteriormente, calculado a partir de cualquier otro diagrama de nudo equivalente de K , difiere solo por un factor de $\pm t^n$ para algún entero n [10].*

El hecho de que el polinomio obtenido pueda diferir por un factor de $\pm t^n$ al calcularse a partir de un diagrama diferente del nudo sugiere que necesitamos alguna forma de normalizar nuestro polinomio para asociar un polinomio único a cada nudo. Una forma posible es establecer $\Delta_K(t) = \pm t^n \Delta_{p,q}(t)$ de modo que el término de menor grado en $\Delta_K(t)$ sea una constante positiva. Esta es la forma normal requerida que nos da nuestro invariante de nudo y se llama el polinomio de Alexander.

Ejemplo 4.1. Cálculo del Polinomio de Alexander Para el nudo trebol.

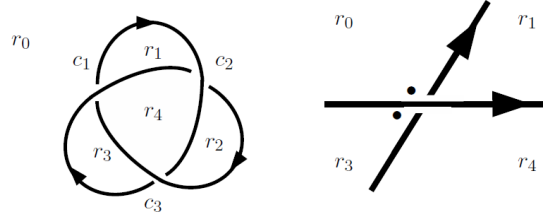


FIGURA 4.3. Tomada de [10]

Consideremos el diagrama del trebol en la figura 4.3. Examinando el cruce c_1 , vemos que las regiones r_0 y r_3 están punteadas y que el orden cíclico en sentido antihorario es r_0, r_3, r_4, r_1 . Esto da como resultado la ecuación:

$$c_1(r) = tr_0 - tr_3 + r_4 - r_1 = 0$$

Repetir el mismo proceso para los puntos de cruce c_2 y c_3 nos da las ecuaciones restantes:

$$c_2(r) = tr_0 - tr_1 + r_4 - r_2 = 0$$

$$c_3(r) = tr_0 - tr_2 + r_4 - r_3 = 0$$

Juntando todo, representamos estas ecuaciones en la matriz:

$$M = \begin{pmatrix} t & -1 & 0 & -t & 1 \\ t & -t & -1 & 0 & 1 \\ t & 0 & -t & -1 & 1 \end{pmatrix}$$

Dos regiones adyacentes son r_3 y r_4 , por lo que eliminamos las dos últimas columnas de la matriz y tomamos el determinante de la matriz cuadrada $M_{3,4}$:

$$\begin{aligned} \Delta_{3,4}(t) = \det(M_{3,4}) &= \begin{vmatrix} t & -1 & 0 \\ t & -t & -1 \\ t & 0 & -t \end{vmatrix} = t \begin{vmatrix} -t & -1 \\ 0 & -t \end{vmatrix} + \begin{vmatrix} t & -1 \\ t & -t \end{vmatrix} \\ &= t^3 - t^2 + t \\ &= t(1 - t + t^2) \end{aligned}$$

Luego sacamos el factor de t para obtener el polinomio normalizado:

$$\Delta_K(t) = 1 - t + t^2$$

Este es el polinomio de Alexander estándar para el nudo trébol y, por el Teorema 4.1, calcular Δ_K a partir de cualquier otro diagrama del nudo trébol dará la misma respuesta.

Desde su descubrimiento, el polinomio de Alexander ha sido una herramienta útil en la investigación de nudos. Con el paso del tiempo, se han desarrollado varios enfoques además del combinatorio utilizado por Alexander, como el enfoque geométrico y el algebraico. En 1969 John Conway demostró que el polinomio de Alexander puede calcularse utilizando solo 2 reglas:

- Regla 1: $\Delta(\bigcirc) = 1$, es decir toma el valor de 1 con el nudo trivial.
- Regla 2: Supongamos que L_+ , L_- y L_0 son los diagramas regulares de los nudos (o enlaces) K_+ , K_- y K_0 respectivamente. Estos diagramas regulares son exactamente los mismos excepto en un punto de cruce. En ese cruce, los diagramas regulares difieren en la manera mostrada en la Figura 4.4. Entonces, los polinomios de estos tres nudos están relacionados a través de:

$$\Delta(L_+) = \Delta(L_-) + z\Delta(L_0).$$

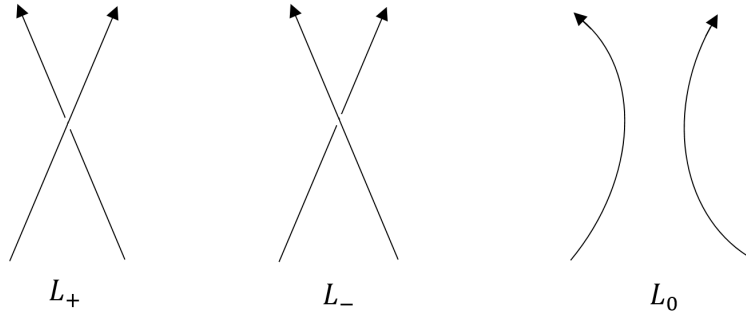


FIGURA 4.4. Diagramas de madejas. Fuente: Elaboración propia.

Los diagramas L_+ , L_- y L_0 son llamados diagramas de madejas y la relación de la regla 2 es llamada la relación de madejas. El polinomio $\Delta(z)$ definido por las reglas anteriores es llamado el polinomio de Conway y su relación con el polinomio de Alexander está dada por el siguiente teorema.

Teorema 4.2. $\Delta_K(t) = \Delta(t^{1/2} - t^{-1/2})$ [4].

En otras palabras, si reemplazamos z por $t^{1/2} - t^{-1/2}$ en el polinomio de Conway, obtenemos el polinomio de Alexander. Por esta razón, el polinomio $\Delta(z)$ se conoce también como el polinomio de Alexander-Conway.

El polinomio de Alexander es bastante útil respecto al problema global de clasificación de nudos. Sin embargo el polinomio de Alexander no nos sirve para demostrar que los tréboles dextrógiro y levógiro no son equivalentes, es decir el polinomio no hace diferencia entre un nudo y su imagen especular.

4.2. Polinomio de Jones. En el año 1984, el matemático neozelandés Vaughan Jones anunció un nuevo polinomio. Jones demostró que su polinomio de Laurent está determinado por dos axiomas, de manera análoga al polinomio de Conway [4].

Definición 4.1. Sea R un anillo conmutativo, un polinomio de Laurent es una expresión de la forma:

$$p(t) = \sum_{k \in \mathbb{Z}} a_k t^k, \quad a_k \in R.$$

El polinomio de Jones que lo denotaremos por V (o como $V_K(t)$ para especificar que es el polinomio de un nudo K en específico) tomará un enlace orientado L y nos devolverá un polinomio que pertenece al anillo de polinomios de Laurent con coeficientes enteros. Este polinomio, de manera similar al polinomio de Alexander, debe cumplir con una relación de madeja.

Definición 4.2. Llamaremos polinomio de Jones a

$$V : L \rightarrow \mathbb{Z}[t^{-1/2}, t^{1/2}]$$

que satisfaga:

- $V(\bigcirc) = 1$, es decir toma el valor de 1 con el nudo trivial.
- $t^{-1}V(L_+) - tV(L_-) + (t^{-1/2} - t^{1/2})V(L_0) = 0$.

Se puede verificar por la definición anterior que se cumplen las siguientes propiedades:

- $V(\bigcirc\bigcirc) = -(t^{-1/2} + t^{1/2})$ y de manera particular para m componentes $V(\bigcirc\bigcirc \dots \bigcirc) = (-1)^{m-1}(t^{-1/2} + t^{1/2})^{m-1}$.
- Sea L un enlace, la unión disjunta de este con el nudo trivial $V(L \sqcup \bigcirc) = -(t^{-1/2} + t^{1/2})V(L)$.

Estas propiedades serán de mucha utilidad a la hora de realizar los cálculos del polinomio.

Ejemplo 4.2. Calculemos el polinomio de Jones para el enlace de Hopf. Usando la definición tenemos que:

$$t^{-1}V(L_+) - tV(L_-) + (t^{-1/2} - t^{1/2})V(L_0) = 0$$

Si nos fijamos en la Figura 4.5 podemos notar que L_+ sería nuestro nudo sin alterar, $L_- \simeq \bigcirc\bigcirc$ y que $L_0 \simeq \bigcirc$, entonces tenemos

$$t^{-1}V(L_+) - tV(\bigcirc\bigcirc) + (t^{-1/2} - t^{1/2})V(\bigcirc) = 0$$

$$t^{-1}V(L_+) + t(t^{-1/2} + t^{1/2}) + (t^{-1/2} - t^{1/2}) = 0$$

$$t^{-1}V(L_+) + t^{3/2} + t^{-1/2} = 0$$

$$V(L_+) = -t^{5/2} - t^{1/2},$$

con lo cual tenemos que el polinomio de Jones del enlace de Hopf es igual a $-t^{5/2} - t^{1/2}$.

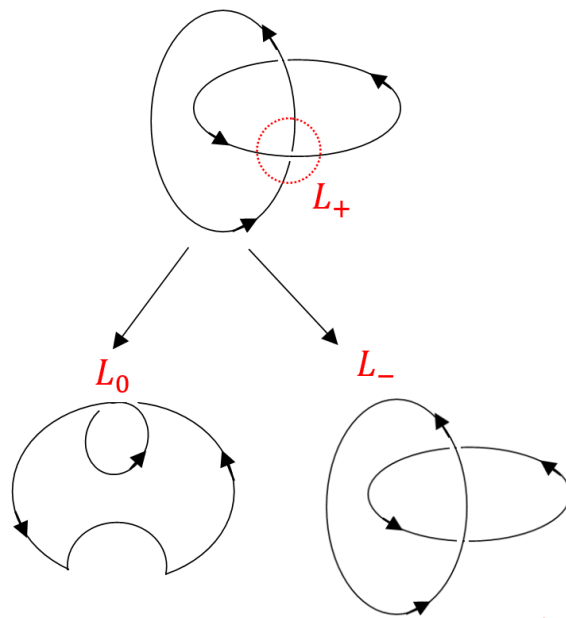


FIGURA 4.5. Diagramas de madejas del enlace de Hopf para el polinomio de Jones . Fuente: Elaboración propia.

Una deficiencia que teníamos con el polinomio de Alexander era que no distinguía entre un nudo y su imagen especular, sin embargo, el polinomio de Jones es bastante eficaz para determinar si un nudo es anfiqueiral o no.

Teorema 4.3. *Sea K^* la imagen especular de K , entonces*

$$V_{K^*}(t) = V_K(t^{-1})[4].$$

Por lo tanto, si un nudo es anfiqueiral, entonces $V_{K^*}(t) = V_K(t^{-1})$, es decir, V es simétrico. En el caso del trébol dextrógiro, se puede comprobar que su polinomio de Jones no es simétrico, por lo tanto no es anfiqueiral. Ahora, ya que el nudo 8 es anfiqueiral se puede comprobar que su polinomio y el de su imagen especular son iguales.

Es natural preguntarse que si el polinomio de Jones de dos nudos es igual, ¿implica esto que los nudos son equivalentes? Sabemos que:

Proposición 4.1. Si $K_1 \simeq K_2$, entonces $V(K_1) = V(K_2)$.

Sin embargo, lo contrario de esta proposición no siempre es cierto; si los polinomios de dos nudos son iguales, no podemos concluir que los nudos sean equivalentes. No obstante, al utilizar la contrapositiva de esta proposición, podemos determinar si dos nudos no son equivalentes.

5. CONCLUSIONES

Luego de investigar la teoría de nudos y examinar algunas de sus invariantes, se presentarán las siguientes conclusiones. Estas conclusiones ofrecen nuevas perspectivas sobre el estudio de los nudos y sugieren caminos prometedores para futuros estudios en este campo.

- En la teoría de nudos, existen otras invariantes además de las presentadas en este trabajo, como la tricolorabilidad, el número de puente, el número de cruces, el número de desanudado, entre otras. Sin embargo, las invariantes más poderosas y las que revolucionaron la clasificación de los nudos fueron las invariantes polinomiales.
- Existen diversas metodologías para la construcción de los polinomios mencionados. Por ejemplo, el polinomio de Alexander puede ser construido no solo mediante la forma combinatoria que hemos explorado, sino también a través de una ruta geométrica utilizando superficies de Seifert, así como mediante la ruta de la topología algebraica mediante el grupo fundamental [10].
- La invariante polinomial más reciente, que podría ser objeto de investigación futura, es el polinomio de HOMFLYPT, desarrollado alrededor del año 1985. Este polinomio generaliza dos polinomios descubiertos previamente: el polinomio de Alexander y el polinomio de Jones. Estos últimos pueden obtenerse mediante sustituciones adecuadas en el polinomio de HOMFLYPT [20].
- El estudio de los polinomios en teoría de nudos sigue muy activo. Actualmente se han desarrollado teorías de homología que, en palabras simples toman la información dada por los polinomios y la categorifican a contextos algebraicos [19] [8].
- Las relaciones de madeja también se pueden llevar a contextos algebraicos usando el concepto de “modulos de madeja” [14].

REFERENCIAS

1. Colin C. Adams, *The Knot Book*, W. H. Freeman and Company, New York, 1994.
2. J. W. Alexander, *Topological invariants of knots and links*, Transactions of the American Mathematical Society, 1928.
3. Lourdes F. Betancort, *Teoría de Nudos*, Universidad de La Laguna, San Cristóbal de La Laguna, 2019.
4. José L. Cisneros, *Introducción a la Teoría de Nudos*, V jornadas de física y matemáticas, Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, 2011.
5. E. Colberg, *A brief history of knot theory*. Disponible en: <https://www.math.ucla.edu/~radko/191.1.05w/erin.pdf> (Accedido en: 19 Febrero 2024).
6. R. Crowell and R. Fox, *Introduction to knot theory*, pringer Science and Business Media, Vol. 57, New York, 2012.
7. Louis H. Kauffman, *New Invariants in the Theory of Knots*, The American Mathematical Monthly, 1998.
8. M. Khovanov, *A categorification of the Jones polynomial*, Duke Mathematical Journal, 2000.
9. C. Kosniowski, *Topología algebraica*, Reverté, Barcelona, 1986.
10. E. Long, *Topological invariants of knots: three routes to the Alexander Polynomial*, Manchester University, Manchester, 2005.
11. James R. Munkres, *Topología*, 2da edición, Prentice Hall, Massachusetts, 2002.
12. Ana E. Orellana, *Teoría de nudos*, Universidad Nacional del Litoral, Santa Fe.
13. Józef H. Przytycki, *Knot theory from Vandermonde to Jones*, Proc. Mexican Nat. Congress Math, Washington, 1991.
14. Józef H. Przytycki, *Skein modules of 3-manifolds*, arXiv preprint math/0611797, 2006.
15. Billy O. Quevedo, *Invariantes en la teoría de nudos*, Universidad de San Carlos de Guatemala, Guatemala.
16. J. Roberts, *Knots Knots*, Lectures from Edinburgh Course Maths, Edimburgo, 1999.
17. Nere S. Rúa, *Teoría de Nudos y el ADN*, Trabajo de fin de grado, Santiago de compostela, 2020.
18. Alexei B. Sossinski, *Knots: mathematics with a twist*, Harvard University Press, Massachusetts, 2002.
19. Gabriel Montoya-Vega, *Una Mirada Inicial a la Teoría de Nudos y a la Homología de Khovanov*, arXiv:2308.10277, 2023.
20. Freyd, Peter and Yetter, David and Hoste, Jim and Lickorish, WB Raymond and Millett, Kenneth and Ocneanu, Adrian, *A new polynomial invariant of knots and links*, 1985.

DEPARTAMENTO DE MATEMÁTICAS, UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS

Dirección de correo electrónico: alanflores@unah.hn

CURVAS ELÍPTICAS, FUNCIONES HASH Y SU USO EN CRIPTOMONEDAS

JAIME YUBINI SIBRIÁN ROMERO

RESUMEN. El blockchain es una de las tecnologías más revolucionarias y modernas del mundo, en palabras simples se dice que es una base de datos distribuida o un libro contable descentralizado y se utiliza en muchos casos, pero uno de los más famosos son las criptomonedas como ser el Bitcoin. En este artículo se dará una breve, pero precisa introducción a los conceptos matemáticos que hay detrás de esta tecnología los cuales son las funciones hash, las curvas elípticas, arboles de Merkle y algunos algoritmos de firma digital.

ABSTRACT. The blockchain is one of the most revolutionary and modern technologies in the world, in simple words it is said to be a distributed database or a decentralized ledger and it is used in many cases, but one of the most famous is Cryptocurrencies such as Bitcoin. This article will give a brief but precise introduction to the mathematical concepts behind this technology, which are Hash functions, elliptic curves, Merkle trees and some digital signature algorithms.

1. INTRODUCCIÓN

Con los avances en la información digital, la tecnología o protocolo blockchain ha emergido como una innovación con el potencial de transformar diversos sectores, desde las finanzas hasta la cadena de suministro y muchos otros más. Gracias a ésta, este nuevo concepto de internet permite compartir valor (como títulos, registros, certificaciones, archivos o canciones) de una forma digital y descentralizada, sin necesidad de una entidad central de confianza que imponga su criterio a los participantes [14]. Todo esto necesita de cierto grado de seguridad y confiabilidad al tratarse de información muy sensible y es donde entra en juego la criptografía. En este artículo se presenta una introducción de los conceptos de blockchain, criptomonedas, Bitcoin y conceptos matemáticos para el desarrollo teórico e implementación del blockchain como son las curvas elípticas y las funciones hash [16]. El contenido de este documento se desarrollará primero dando una idea general al blockchain y criptomonedas, luego se introducen conceptos matemáticos preliminares para entender la teoría de curvas elípticas y funciones hash y así poder entender y construir la idea del Bitcoin.

Los algoritmos que utilizamos en criptografía aplicando curvas elípticas y las funciones hash son algoritmos muy robustos que pueden soportar hasta ciertos ataques cuánticos lo cual hacen que el protocolo blockchain sea algo muy confiable, también utiliza algo llamado “Proof of Work” la cual consiste en realizar cálculos matemáticos

Fecha: February 19, 2024.

Palabras y frases clave. Curvas Elípticas,criptografía,criptografía asimétrica,sistemas criptograficos.

en cierto tiempo para verificar la autenticidad de las transacciones del blockchain [15].

La adopción de criptomonedas respaldadas por blockchain se presenta como una oportunidad ineludible para Honduras. Enfocándonos en el ámbito financiero, reconocemos que esta tecnología no solo es una tendencia global, sino una herramienta transformadora que puede abordar desafíos en nuestra sociedad, como la corrupción y la malversación de fondos [14].

Resulta importante conocer los fundamentos matemáticos que sustentan el blockchain para comprender plenamente su potencial. Esto nos capacita no solo para adaptarnos a las criptomonedas, sino también para explorar aplicaciones más amplias, una de ellas es la administración pública, donde la simplificación y agilización de procesos pueden ser la clave para mitigar la corrupción y garantizar la transparencia en la gestión de registros.

La Universidad Nacional Autónoma de Honduras, como institución académica líder en Honduras, ha delineado claramente sus prioridades de investigación para los próximos años. Este trabajo sigue la línea de investigación de uno de los ejes prioritarios planteados por la UNAH, dedicado a Desarrollo Económico y Social, este eje tiene tres temas prioritarios, pero nos centraremos en el (c) Globalización, productividad y competitividad [17]. Reconocemos la importancia estratégica de este enfoque para el desarrollo económico y social de Honduras.

2. ANTECEDENTES

En el blockchain, algo muy fundamental es la criptografía, ya que se utilizan algunas técnicas criptográficas como el cifrado asimétrico (clave pública) y la función hash [16].

Una función hash es un procedimiento que convierte cualquier conjunto arbitrario de datos en una serie de caracteres de longitud fija, sin importar el tamaño de los datos de entrada [16]. El resultado obtenido se conoce como hash, resumen, digest o imagen. Con frecuencia, el término “hash” se emplea tanto para hacer referencia a la función hash como al valor generado al aplicar dicha función sobre un mensaje específico.

La aparición de la primera función hash fue en el año 1961 [4]. Wesley Peterson creó la función CRC (Cyclic Redundancy Check), creada para comprobar cómo de correctos eran los datos transmitidos en redes (como Internet) y en sistema de almacenamiento digital.

El cifrado de clave pública es un método de encriptación que emplea un par de claves: una clave pública para cifrar los datos y una clave privada para descifrarlos. Estas claves están vinculadas entre sí. El protocolo de clave pública fue propuesto en 1976 por Whitfield Diffie y Martin Hellman [5].

La criptografía de curva elíptica constituye un tipo de criptografía asimétrica o de clave pública que se fundamenta en el problema del logaritmo discreto, expresado a través de operaciones de suma y multiplicación de puntos en una curva elíptica

[1]. Esta técnica se emplea para firmar transacciones y verificar la propiedad de las direcciones de las criptomonedas en el blockchain.

En la década de 1980, Miller [12] y Koblitz [11] introdujeron las curvas elípticas en el campo de la criptografía, mientras que Lenstra demostró su utilidad en la factorización de números enteros. Una de sus ventajas destacadas es su capacidad para proporcionar un nivel de seguridad equiparable a los criptosistemas clásicos, incluso con tamaños de clave más reducidos.

No nos inspira confianza depender en gran medida de un sistema centralizado para almacenar nuestra información confidencial con el riesgo de robo, corrupción o modificación. Stuart Haber y W. Scott Stornetta propuso el concepto de blockchain en su artículo de 1991 “Cómo marcar el tiempo de un documento digital” [7]. En él, proponen una solución práctica para garantizar que la fecha de la firma original y el documento digital que se firmó en una fecha específica no pueden ser alterados. La solución recomendada guarda los documentos y su sello de fecha en una cadena de bloques.

En 2007 se produjo un periodo de inestabilidad económica debido a la dinámica especulativa llevada a cabo en el sector inmobiliario a lo largo de la primera década del siglo XXI [6]. Esto llevó a una parte de la población a cuestionar el sistema financiero del momento y a que en ciertos sectores sociales se generara desconfianza respecto a los bancos centrales, de forma que algunos colectivos reaccionaron planteando modelos financieros alternativos orientados a eliminar la dependencia del sistema monetario tradicional.

Por lo tanto, se crea un entorno favorable para la implementación de algo conocido como blockchain, y a finales del 2008, una persona o grupo de personas publicaron un documento que describía un sistema electrónico de pago descentralizado entre pares llamado Bitcoin: “A Peer-to-Peer Electronic Cash System” [3]. Aún no se ha descubierto quién es Satoshi Nakamoto, el seudónimo bajo el cual se publicó el documento. Como resultado, Bitcoin se convirtió en la primera aplicación popular de la tecnología blockchain.

Desde la creación de Bitcoin, muchas nuevas criptomonedas han surgido en el mercado, una de ellas es el Ethereum, que actualmente es la segunda mayor divisa de criptomonedas en términos de capitalización de mercado. A la edad de 19 años, Vitalik Buterin presentó la idea de Ethereum. Ese mismo año, publicó un “whitepaper” que decía que Ethereum era “la próxima generación de contratos inteligentes y plataforma de aplicaciones descentralizadas” [19]. Frontier, la primera versión de Ethereum (Ethereum 1.0), se lanzó el 30 de julio de 2015. Tenía las siguientes funciones: permite a los usuarios ejecutar contratos inteligentes y extraer Ether.

En esta sección se estudió un poco de la historia de los conceptos fundamentales y algunas aplicaciones del blockchain para así poder entender de dónde se derivan estos y poder entender mejor el funcionamiento del blockchain. En la siguiente sección se estudian estos conceptos con mayor profundidad.

3. CURVAS ELÍPTICAS, FUNCIONES HASH Y SU USO EN CRIPTOMONEDAS

3.1 Conceptos básicos.

3.1.1 Definición del blockchain. El blockchain es una base de datos distribuida entre varios participantes, protegida criptográficamente y organizada matemáticamente en bloques de transacciones. En pocas palabras, es una base de datos descentralizada inalterable. El hecho de que sea un sistema que permita que partes que no confían plenamente unas en otras puedan llegar a un consenso sobre la existencia, el estado y la evolución de una serie de factores compartidos es otro elemento muy importante a tener en cuenta en ella. El consenso es fundamental para un sistema blockchain porque es la base sobre la cual todos los participantes pueden confiar en la información que se encuentra allí [14].

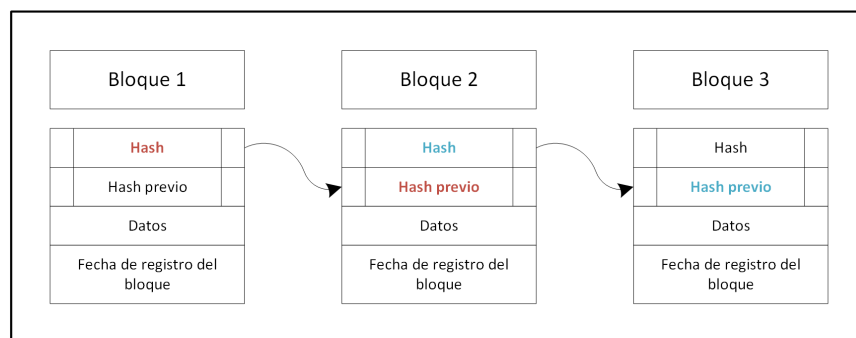


FIGURE 1. Esquema básico de un blockchain [2]

3.1.2 Elementos de la cadena de bloques (blockchain).

- (a) Bloques: conjunto de transacciones confirmadas y validadas por los mineros.
- (b) Mineros: Los mineros son ordenadores/chips dedicados que aportan poder computacional a la red para verificar las transacciones que se llevan a cabo.
- (c) Nodos: Un nodo es un ordenador/chip conectado a la red utilizando un software que almacena y distribuye una copia actualizada en tiempo real de la cadena de bloques. Cada vez que un bloque se confirma y se añade a la cadena se comunica a todos los nodos y este se añade a la copia que cada uno almacena.

3.1.3 Criptomonedas. Una criptomoneda, criptodivisa o criptoactivo puede definirse como una representación digital de valor, no emitida por un banco central, una entidad de crédito o una entidad de dinero electrónico, que, en algunas circunstancias, puede utilizarse como alternativa al dinero tradicional [2].

3.1.3 Proof of Work. La Prueba de Trabajo es un algoritmo que requiere que los participantes de una red criptográfica realicen un trabajo computacionalmente costoso para validar y procesar transacciones, así como para agregar nuevos bloques al blockchain [9].

3.2 Preliminares.

3.2.1 Función hash.

Definición 3.2.1.1. Una función $f : X \rightarrow Y$ es unidireccional si $\forall x \in X$, calcular $f(x) = y$ es poco costoso computacionalmente y, por otro lado, sea $y \in f(X)$, es muy costoso computacionalmente encontrar $x \in X$ tal que $f(x) = y$. A esta última característica se le llama resistencia a la preimagen.

Definición 3.2.1.2. Una función $f : X \rightarrow Y$ es resistente a las colisiones si es computacionalmente difícil encontrar dos elementos de X , x_1 y x_2 , tal que $x_1 \neq x_2$ y $f(x_1) = f(x_2)$.

Se denomina función hash a una función unidireccional g resistente a las colisiones y muy sensible a las modificaciones, capaz de transformar una información o mensaje de longitud arbitraria en otro que tiene un tamaño fijado de antemano [16]. Las funciones hash más notables son los algoritmos hash seguros (SHA), RIPEMD-160.

3.2.2 Árboles de Merkle. Un árbol Merkle, también conocido como árbol hash binario, es una estructura de datos que se utiliza para resumir y verificar de manera eficiente la integridad de grandes conjuntos de datos. Los árboles Merkle son árboles binarios que contienen hashes criptográficos. El término “árbol” se utiliza en informática para describir una estructura de datos ramificada, pero estos árboles generalmente se muestran al revés con la “raíz” en la parte superior y las “hojas” en la parte inferior de un diagrama, como verá en el ejemplo que sigue [1].

Ejemplo 3.2.2.1. Si se quiere generar un árbol de Merkle asociado a 4 transacciones A,B,C,D mediante una función hash H, este tendría la forma que se muestra en la Figura 2.

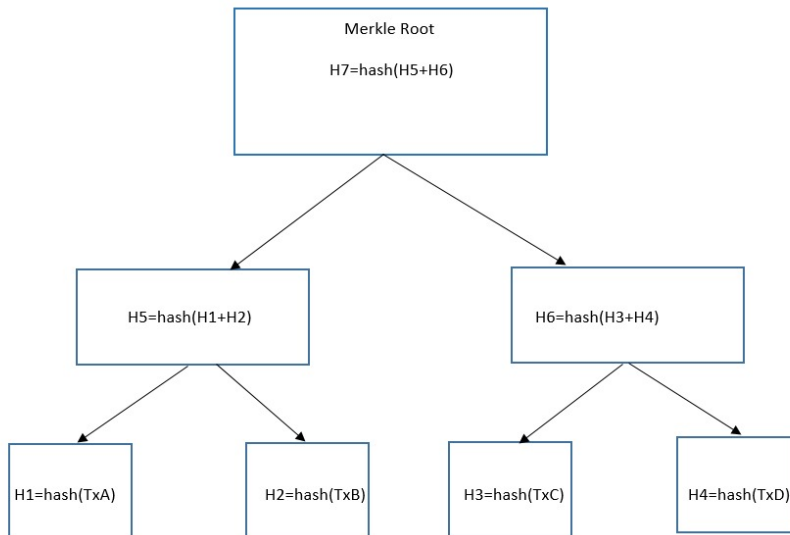


FIGURE 2. Fuente: Elaboración propia

3.2.3 Curva Elíptica.

Definición 3.2.3.1. Una curva elíptica sobre un campo K es una curva algebraica que esta dada por una ecuación del tipo: $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_5$, con $a_1, a_2, a_3, a_4, a_5 \in K$ y $\Delta \neq 0$, llamada ecuación general de Weierstrass donde Δ es el discriminante de la ecuación, donde:

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6,$$

$$d_2 = a_1^2 + 4a_2$$

$$d_4 = 2a_4 + a_1a_3$$

$$d_6 = a_3^2 + 4a_6$$

$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

Estamos interesados en los campos de números racionales Q , números reales R , números complejos C y el de los números enteros modulo p es decir $F_q = Z/pZ$. Si la característica de un campo F_q es distinta de 2 y 3, usando transformaciones lineales en la ecuación general de Weierstrass, la ecuación de la curva se puede reducir a:

$$y^2 = x^3 + ax + b, a, b \in F_q$$

Definición 3.2.3.2. Si el polinomio $x^3 + ax + b$ tiene raíces repetidas, decimos que la curva elíptica $y^2 = x^3 + ax + b$ es singular. En caso contrario decimos que la curva elíptica es no singular. Una curva es singular si y solo si su discriminante $\Delta = 16(4a^3 + 27b^2) = 0$

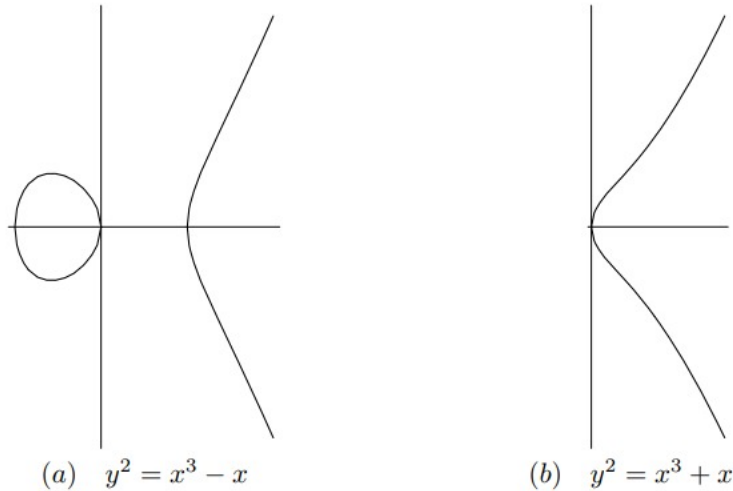


FIGURE 3. Representación geométrica de algunas curvas elípticas [18].

3.2.3.1 Curva Elíptica como estructura de grupo. Algo muy interesante de las curvas elípticas es que se puede definir sobre ellas una operación $+$ de modo que $(E, +)$ sea un grupo conmutativo con elemento identidad O .

Para poder realizar la suma de dos puntos se debe realizar lo siguientes pasos:

(1) sea $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$ dos puntos distintos sobre una curva elíptica

$$E : y^2 = x^3 + ax + b.$$

- (2) Trazamos una recta que pasa por los puntos $P1$ y $P2$, asegurando que la recta L interseca E en un tercer punto $P3$. Suponemos que la recta L es de la forma $y = mx + b$ y que no es vertical (Se considerará posteriormente dicha situación).
- (3) La intersección entre L y E son las soluciones al sistema, lo que es equivalente a resolver la ecuación cubica. Sin embargo, conocemos dos raíces $x = x_1$ y $x = x_2$, por lo que se tiene garantizado el tercer punto $P3$.
- (4) Si $P3 = (X_3, Y_3) \in E$ entonces $-P3 = (X_3, -Y_3)$.

Teorema 3.1. Sea E una curva elíptica definida sobre K , donde K es un campo, entonces para $P1, P2, P3$ puntos sobre E , se cumple que:

- (1) $P1 + P2 = P2 + P1$ - Ley Conmutativa.
- (2) $(P1 + P2) + P3 = P1 + (P2 + P3)$ - Ley Asociativa.
- (3) $P + \infty = \infty + P = P$ - El punto ∞ es identidad.
- (4) $P + (-P) = (-P) + P = \infty$ - El punto inverso de P .

En otras palabras, los puntos en E forman un grupo abeliano aditivo con ∞ como elemento identidad.

Nota: En el caso de la curva elíptica, se define un punto en el infinito (∞) como el punto que corresponde a la intersección de todas las tangentes a la curva que no tienen otros puntos de intersección con ella.

Teorema 3.2. (Ley de suma). Sea $P1 = (x_1, y_1)$ y $P2 = (x_2, y_2)$ puntos sobre la curva elíptica $E : y^2 = x^3 + ax + b$. Entonces $P1 + P2 = (x_3, y_3)$ donde $x_3 = m^2 - x_1 - x_2$, $y_3 = -m(x_3 - x_1) - y_1$ y

$$m = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & : P1 \neq P2 \\ \frac{3x_1^2 + a}{2y_1} & : P1 = P2 \end{cases}$$

si m es infinito, entonces $P1 + P2 = \infty$.

Ejemplo 3.2.3.1. Sea $E(F_7)$ una curva elíptica definida por la ecuación $y^2 = x^3 - x + 1$ y la recta que pasa por los puntos P y Q está dada por $y = 3x + 1$. Sean $P = (0, 1)$ y $Q = (2, 0)$ puntos de la curva. Calcular $P + Q$.

Primero vamos a realizar la intersección de las dos ecuaciones.

$$x^3 - x + 1 = (3x + 1)^2 = 9x^2 + 6x + 1,$$

luego

$$\begin{aligned} x^3 - x &= 2x^2 + 6x \\ x^3 - 2x^2 - 7x &= 0 \\ x^3 - 2x^2 &= 0 \\ x^2(x - 2) &= 0 \end{aligned}$$

Las soluciones de esta ecuación son 0 y 2.

Por lo tanto,

$$P + Q = P = (0, 1) = (0, 6)$$

3.2.3.2 Curva Elíptica módulo un primo. Las curvas elípticas sobre Z_p , se define de igual manera que en los reales al igual que la operación de suma. En general, en los criptosistemas basados en curvas elípticas se utiliza un cuerpo finito primo F_p .

Definición 3.2.3.3. Sea $p > 3$ un número primo, La curva elíptica $E : y^2 = x^3 + ax + b$ sobre Z_p es el conjunto de soluciones $(x, y) \in Z_p \times Z_p$, donde $a, b \in Z_p$ son constantes tal que $4a^3 + 27b^2 \neq 0 \pmod{p}$ junto con un punto especial O llamado punto al infinito.

$$E(F_p) = \{(x, y) \in Z_p^2 | y^2 = x^3 + ax + b\} \cup \{O\}$$

Ejemplo 3.2.3.2. Sea E una curva elíptica definida por la siguiente ecuación $y^2 = x^3 + 4x + 4 \pmod{2773}$ y $P = (1, 3)$. Calcular $P + P$.

Utilizando el teorema 3.2 sabemos que si $P_1 = P_2$ entonces $m = \frac{3x_1^2 + a}{2y_2}$ por lo cual:

$$m \equiv \frac{3(1)^2 + 4}{2(3)} \equiv \frac{7}{6} \equiv 7 \times 2311 \equiv 2312 \pmod{2773}$$

Luego,

$$\begin{aligned} x_3 &\equiv m^2 - x_1 - x_2 \equiv (2312)^2 - 1 - 1 \pmod{2773} \equiv 1771 \pmod{2773} \\ y_3 &\equiv m(x_3 - x_1) - y_1 \equiv 2312(1 - 1771) - 3 \equiv 705 \pmod{2773}. \end{aligned}$$

Por lo tanto,

$$P + P = (1771, 705).$$

Definición 3.2.3.4. (Número de puntos en la curva elíptica)

Sea $E : y^2 = x^3 + ax + b \pmod{p}$ una curva elíptica, donde $p \geq 5$ es primo. Podemos enumerar los puntos en E como $x = 0, 1, \dots, p - 1$ y viendo cuando $x^3 + ax + b$ es un cuadrado mod p .

La dificultad de encontrar los puntos de una curva elíptica aumenta conforme lo hace el orden del cuerpo finito sobre el que está definida la curva. Pero el teorema de Hasse proporciona una cota para el orden(cardinal) de una curva elíptica sobre un cuerpo finito.

Teorema 3.3 (Hasse). Sea E una curva elíptica sobre un campo finito F_q . Se tiene la siguiente cota:

$$|\#E - q - 1| \leq 2\sqrt{q}$$

De esta forma, el número de puntos de E ($\#E$) puede acotarse de la siguiente manera:

$$q + 1 - 2\sqrt{q} \leq \#E \leq q + 1 + 2\sqrt{q}$$

Para trabajar en criptografía con curvas elípticas, se necesita usar un grupo finito, a fin de poder computarse en un tiempo razonable.

3.2.4 Firma digital.

Una firma digital es un esquema criptográfico que garantiza la autenticidad e integridad de un mensaje o documento digital, así como la no repudiación por parte del firmante. Esto se logra mediante el uso de algoritmos criptográficos que generan una firma única y verificable, basada en la clave privada del firmante, y que puede ser verificada por cualquiera que tenga acceso a la clave pública del firmante[10]. La firma digital emplea criptografía asimétrica o de clave pública, que se caracteriza por el uso de dos claves relacionadas pero distintas para cifrar y descifrar información. Esta distinción la separa de la criptografía simétrica, donde se utiliza una única clave para ambas funciones. El par de claves incluye una clave pública, compartida con los usuarios con los que se desea comunicarse, y una clave privada, que debe mantenerse en secreto y ser conocida solo por su dueño.

Actualmente, los criptosistemas asimétricos se basan en la dificultad de resolver problemas matemáticos específicos, como el problema del logaritmo discreto.

3.2.4.1 Problema del logaritmo discreto.

Definición 3.2.4.1. Sea p un primo y sean a, b números enteros distintos de cero mod p . Supongamos que sabemos que existe un número entero k tal que:

$$a^k \equiv b \pmod{p}.$$

El problema de encontrar k se llama problema de logaritmo discreto. Si n es el entero positivo más pequeño tal que $a^n \equiv 1 \pmod{p}$, podemos suponer que $0 \leq x < n$, y luego denotamos como $k = L_a(b)$ y llámelo el log discreto de b con respecto a a .

Definición 3.2.4.2 (Problema del logaritmo discreto). Sea G cualquier grupo, escrito multiplicativamente por el momento, y sean $a, b \in G$. Supongamos que sabemos que $a^k = b$ para algún entero k . En este contexto, el problema del logaritmo discreto consiste nuevamente en encontrar k .

Ejemplo 3.2.3.1. Sea $p = 17$ y $a = 3$ es una raíz primitiva. Resolver la siguiente expresión $3^k = 7 \pmod{17}$.

Por inspección:

$$3^1 \equiv 3, 3^2 \equiv 9, 3^3 \equiv 10, 3^4 \equiv 13, 3^5 \equiv 5, 3^6 \equiv 15, 3^7 \equiv 11, 3^8 \equiv 16, 3^9 \equiv 14, 3^{10} \equiv 8, 3^{11} \equiv 7$$

Por lo tanto la solución es $k = 11$

El problema del logaritmo discreto en curva elíptica. Neal Koblitz y Victor Miller propuso en 1985 el uso de un grupo aditivo de curva elíptica. Un sistema de cifrado asimétrico se define de manera similar a otros sistemas asimétricos considerando la estructura definida por los puntos de una curva elíptica [11]. Sin embargo, hay algunas diferencias significativas entre ambos tipos de sistemas: en un sistema basado en curvas elípticas, la longitud de las claves es significativamente menor, pero se alcanza un nivel de seguridad equivalente. Además, la operación de suma que se realiza con los puntos de una curva elíptica es más complicada.

Definición 3.2.4.3 (Problema del logaritmo discreto con curvas elípticas). Sea E una curva elíptica sobre un cuerpo finito F_q , un punto $P \in E(F_q)$ de orden n y un punto $Q \in \langle P \rangle$, encontrar el entero $l \in [0, n - 1]$ tal que $Q = lP$. El entero l es llamado el logaritmo discreto de Q en base P .

Para elegir una curva elíptica E sobre F_p para uso criptográfico, hay que seleccionar un número primo p grande tal que el orden de la curva sea un número primo o, en su defecto, el producto de un primo por un entero pequeño [8].

3.2.4.2 Algoritmo de firma digital con curva elíptica. El algoritmo de firma digital con curva elíptica o ECDSA requiere una curva elíptica sobre un cuerpo finito, así como una clave privada y pública del firmante (U, u) , donde u es un entero positivo (clave privada) y $U = uP$ es un punto de la curva E (clave pública). Sabemos que los protocolos de firma digital está formado por dos algoritmos diferentes, el de elaboración de firma y el de verificación de la misma[18].

Algoritmo 1 Elaboración de firma con curva elíptica

```
1: Elegir una curva elíptica  $E$  definida sobre un campo finito  $Fp$ .
2: Elegir un punto base  $P$  en la curva  $E$  con un orden grande  $n$ .
3:  $M \leftarrow f(m)$  ▷ Se calcula la función hash del mensaje  $m$ 
4: hacer
5:    $k \leftarrow (0, n - 1)$  ▷ Se toma un entero  $k$  tal que  $0 \leq k \leq n - 1$ 
6:    $Q \leftarrow kP$  ▷  $Q = (x_0, y_0)$ 
7:    $r \leftarrow x_0 \pmod{n}$ 
8: mientras  $r = 0$ 
9:    $s \leftarrow k^{-1}(M + ur) \pmod{n}$ 
10: retorna  $(r, s)$  ▷ El par  $(r, s)$  es la firma
```

Algoritmo 2 Verificación de firma con curva elíptica

```
1: Elegir una curva elíptica  $E$  definida sobre un campo finito  $Fp$ .
2: Elegir un punto base  $P$  en la curva  $E$  con un orden grande  $n$ .
3:  $M \leftarrow f(m)$  ▷ Se calcula la función hash del mensaje  $m$ 
4: si  $r \notin [0, n - 1] \cap Z$  o  $s \notin [0, n - 1] \cap Z$ 
5:   retorna inválida
6:  $z_1 \leftarrow Ms^{-1} \pmod{n}$ 
7:  $z_2 \leftarrow rs^{-1} \pmod{n}$ 
8:  $(x_1, y_1) \leftarrow z_1P + z_2U$ 
9: si  $r \equiv x_1 \pmod{n}$ 
10:  retorna inválida
11: sino
12:  retorna inválida
```

3.3 Funcionamiento general.

En esta sección, vamos a construir la idea de Bitcoin con todas las ideas previamente definidas, ver cómo funciona en general e implementar los algoritmos de la criptomoneda.

3.3.1 Bitcoin. Bitcoin es una colección de conceptos y tecnologías que forman la base de un ecosistema de dinero digital. Las unidades monetarias llamadas bitcoins se utilizan para almacenar y transmitir valor entre los participantes de la red bitcoin [1].

3.3.2 Blockchain de Bitcoin.

La comprensión de la estructura de un bloque en la cadena de Bitcoin es esencial para comprender cómo funciona la criptomoneda.

Estructura del Bloque de Bitcoin

- Versión (Version): Un número que indica la versión del bloque.
- Hash del bloque anterior (Previous Block Hash): La referencia al hash del bloque anterior en la cadena, lo que establece el orden cronológico de los bloques.

- Raíz del árbol de Merkle (Merkle Root): El hash de la raíz de un árbol de Merkle que se utiliza para resumir todas las transacciones incluidas en el bloque.
- Marca de tiempo (Timestamp): El momento en que el bloque fue creado, representado como un sello de tiempo Unix.
- NBits: El objetivo de dificultad actual codificado en un formato compacto.
- Nonce: Un número aleatorio que se modifica durante el proceso de minería para encontrar un hash del bloque que cumpla con el requisito de dificultad.

3.3.3 Funcionamiento del Bitcoin. La transacción que realiza un usuario se transmite a todos los nodos mineros de la red. Cada nodo minero crea un bloque agrupando transacciones después de verificar la validez de sus firmas y verificar la suficiente liquidez comprobando los bloques previos e intentando resolver la prueba de trabajo del bloque. Un nodo transmite el bloque a todos los nodos después de resolver esta prueba. Todos los nodos verifican el resultado de la prueba de trabajo y aceptan el bloque si todas las transacciones en él son válidas. En la creación de su siguiente bloque, los nodos mineros utilizan el hash de su cabecera como hash previo para demostrar que han aceptado el bloque.

Bitcoin utiliza un sistema criptográfico asimétrico de firma digital basado en curva elíptica para verificar las transacciones entre usuarios. Se utiliza la función hash SHA-256 para obtener los resúmenes de los bloques que conforman su blockchain, para los árboles de Merkle y para el algoritmo de firma digital. Para las direcciones se utiliza la función hash RIPEMD-160 [1].

3.3.3.1 Implementación de la firma digital del Bitcoin.

El algoritmo de firma digital con curva elíptica descrito anteriormente se utiliza en Bitcoin, utilizando la función hash SHA-256. El algoritmo mencionado utiliza la curva elíptica secp256k1 y tiene los siguientes parámetros.

- 1) $a = 0; b = 7$ donde $y^2 = x^3 + ax + b = x^3 + 7$
- 2) $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 2^0$
- 3) $n = 115792089237316195423570985008687907852837564279074904382605163141518161494337$
- 4) $G = (x_0, y_0)$
 $x_0 = 55066263022277343669578718895168534326250603453777594175500187360389116729240$
 $y_0 = 32670510020758816978083085130507043184471273380659243275938904335757337482424$

En donde a, b son parámetros de la curva elíptica y G es un punto base que pertenece a la curva el cual genera el grupo cíclico de orden n .

Para la implementación del algoritmo de firma digital con curvas elípticas se utilizó el lenguaje de programación Python.

```

1 import hashlib
2 import random
3
4 """ la curva secp256k1, comunmente utilizada en Bitcoin y^2 = x^3 + 7
   """

```

```

5 p = pow(2, 256) - pow(2, 32) - pow(2, 9) - pow(2, 8) - pow(2, 7) - pow
  (2, 6) - pow(2, 4) - pow(2, 0) #primo p
6 n = 11579208923731619542357098500868790
  7852837564279074904382605163141518161494337 #Orden
7 a = 0 ;b = 7
8 G=(55066263022277343669578718895168534326250603453777594175500187
  360389116729240,3267051002075881697808308513050704318447127338065
  9243275938904335757337482424)
9
10 # Funcion para sumar dos puntos en la curva
11 def Suma_puntos(p1, p2,p):
12
13     if p1 == "identidad":
14         return p2
15     elif p2 == "identidad":
16         return p1
17     elif p1[0] == p2[0] and p1[1] != p2[1]:
18         return "identidad"
19     else:
20         if p1[0] == p2[0] and p1[1] == p2[1]:
21             beta = (3*p1[0]*p2[0] + a)*pow(2*p1[1], -1, p)
22
23         else:
24             beta = (p2[1]-p1[1])*pow(p2[0]-p1[0], -1, p)
25
26         x3 = (beta*beta -p1[0]-p2[0]) % p
27         y3 = (beta* (p1[0]-x3)-p1[1]) % p
28
29         punto_en_la_curva((x3, y3), p, a, b)
30         return x3,y3
31
32 # Funcion para multiplicar un punto por un escalar.
33 def multiplicacion_escalar(escalar, punto, p):
34     r = "identidad"
35     temp = punto
36     while escalar != 0:
37         if escalar % 2 == 1:
38             r = Suma_puntos(r, temp, p)
39             temp = Suma_puntos(temp, temp, p)
40             escalar = escalar // 2
41     return r
42 #Funcion para ver si el punto esta en la curva
43 def punto_en_la_curva(P,p,a,b):
44     x,y=P
45     assert (y*y) % p == (pow(x, 3,p) + a*x+ b) % p
46
47 # Funcion para generar un par de claves publica y privada
48 def generador_claves():
49     private_key = random.randint(1, n - 1)
50     public_key = multiplicacion_escalar(private_key, G,p)
51     return private_key, public_key,p
52
53 # Elaboracion de firma

```

```

54 def Firmar(message, private_key,p):
55     z = int.from_bytes(hashlib.sha256(message).digest(), "big")
56     k = random.randint(1, n - 1)
57     r, y = multiplicacion_escalar(k, G,p)
58     s = pow(k, -1, n) * (z + r * private_key) % n
59     return r, s
60
61 # Verificacion de una firma
62 def verify(message, firma, public_key,p):
63     r, s = firma
64     if not (0 < r < n and 0 < s < n):
65         return False
66     z = int.from_bytes(hashlib.sha256(message).digest(), "big")
67     w = pow(s, -1, n)
68     u1 = (z * w) % n
69     u2 = (r * w) % n
70     x, y = Suma_puntos(multiplicacion_escalar(u1, G,p),
71                       multiplicacion_escalar(u2, public_key,p),p)
72     return r == x % n
73
74 private_key, public_key,k = generador_claves()
75 mensaje = b"UNAH!"
76 firma = Firmar(mensaje, private_key,p)
77 print("firma:", firma)
78 print("Verificacion:", verify(mensaje, firma, public_key,p))

```

En el código anterior primero se elige la curva elíptica y luego se programó la función suma de puntos que consiste básicamente en sumar dos puntos de la curva elíptica donde la suma está definida en el teorema 3.2, ya que sabemos que la suma con la curva elíptica forma un grupo abeliano. Luego se define la función producto por escalar que de igual forma, ya que multiplicar por un escalar n resulta sumar ese punto n -veces.

Luego se define una función que genera una clave privada y una clave pública para posteriormente utilizarse en el algoritmo criptográfico de firma digital. Luego se crea la función de elaboración y verificación de firma digital que, como dice su nombre, elabora una firma digital y luego la verifica basándose en los algoritmos previamente definidos en este documento.

3.3.3.2 Árboles de Merkle en Bitcoin. En Bitcoin, los árboles Merkle resumen todas las transacciones en un bloque, lo que crea una huella digital general de todo el conjunto de transacciones, lo que ofrece un proceso muy efectivo para determinar si una transacción está incluida en un bloque. Un árbol de Merkle se construye haciendo un hash recursivo de varios nodos hasta que solo hay un hash, conocido como la raíz o la raíz de Merkle. Los árboles Merkle de Bitcoin utilizan SHA256 aplicado dos veces, también conocido como doble-SHA256.

La implementación del árbol de Merkle hecho en lenguaje de programación Python [13].

4. CONCLUSIONES.

La criptografía de curvas elípticas se basa en propiedades matemáticas definidas sobre un campo finito. Las operaciones de suma y multiplicación por escalares se definen sobre estas curvas, donde la suma es un grupo abeliano, lo cual es necesario para poder implementar los algoritmos criptográficos y brindar la robustez y confiabilidad de estos.

Las curvas elípticas ofrecen un nivel de seguridad alto con respecto a otros métodos criptográficos y se basan en la resolución del problema del logaritmo discreto, lo cual lo hace un método muy confiable. Es por eso que se utiliza en el blockchain, pues al realizarse transacciones que representan valor en tiempo real, deben ser métodos muy seguros, lo cual hace que Bitcoin sea un sistema muy seguro y confiable.

Al tratarse de funciones que son difíciles de trabajar computacionalmente, las funciones hash son esenciales para garantizar que los datos del blockchain sean infalibles y protegidos de ataques de modificación. También se utilizan para brindar privacidad porque los usuarios o direcciones de Bitcoin son anónimos.

El blockchain es una tecnología fascinante que ha ganado popularidad con el tiempo. Puede usarse en una variedad de campos del desarrollo humano y su base matemática es muy confiable.

5. TRABAJOS A FUTURO.

- (1) Poder aplicar la criptografía de curvas elípticas a sistemas de información que beneficien al país.
- (2) Debido al sistema médico obsoleto e ineficiente del país, podemos analizar y desarrollar un sistema de historiales médicos descentralizado y globalizado utilizando curvas elípticas y funciones hash.
- (3) Se espera que se sigan desarrollando curvas elípticas más seguras y eficientes en el futuro para mejorar la seguridad de las criptomonedas.

REFERENCES

- [1] Andreas M Antonopoulos. *Mastering Bitcoin: Programming the open blockchain.* " O'Reilly Media, Inc.", 2017.
- [2] EUROPEAN CENTRAL BANK. *Virtual currency schemes – a further analysis.* 2015.
- [3] Nakamoto S Bitcoin. *Bitcoin: A peer-to-peer electronic cash system.* 2008.
- [4] Juan Pablo Claros Romero et al. *Aplicación de Blockchain para el uso de transportes.* 2021.
- [5] Angel del Río Mateos. *Introducción a la criptografía.* 2021.
- [6] David Arroyo Guardado, Jesús Díaz Vico, and Luis Hernández Encinas. *Blockchain.* CSIC, 2019.
- [7] Stuart Haber and W Scott Stornetta. *How to time-stamp a digital document.* Springer, 1991.
- [8] Darrel Hankerson and Alfred Menezes. *Elliptic curve cryptography.* Springer, 2021.
- [9] Hengrui Jia, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. *Proof-of-learning: Definitions and practice.* 2021.
- [10] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography: principles and protocols.* Chapman and hall/CRC, 2007.
- [11] Neal Koblitz. *Elliptic curve cryptosystems*, volume 48. 1987.
- [12] Victor S Miller. *Use of elliptic curves in cryptography.* 1985.
- [13] onuratakan. mixmerkle tree. https://github.com/onuratakan/mix_merkletree, 2022.
- [14] Alexander Preukschat. *Blockchain: la revolución industrial de internet.* Gestión 2000, 2017.
- [15] Sothearith Seang, Dominique Torre, et al. *Proof of Work and Proof of Stake consensus protocols: a blockchain application for local complementary currencies*, volume 3. 2018.
- [16] Wade Trappe. *Introduction to cryptography with coding theory.* Pearson Education India, 2006.
- [17] UNAH. *Prioridades de investigación 2015-2019 ejes y temas prioritarios.*
- [18] Lawrence C Washington. *Elliptic curves: number theory and cryptography.* CRC press, 2008.
- [19] Wisdomtree. *Ethereum's history: from zero to 2.0.* 2021.

Dirección actual: Departamento de Ciencias de la Universidad Nacional Autónoma de Honduras

Dirección de correo electrónico: jaime.sibrian@unah.hn

ANÁLISIS PROSPECTIVO DE LA PRECIPITACIÓN PLUVIAL EN HONDURAS MEDIANTE MODELOS ESTADÍSTICOS Y MACHINE LEARNING

ANTHONY JAVIER SANCHEZ ROMERO

RESUMEN. This study addresses the challenge of forecasting precipitation in Honduras through a time series analysis, using data provided by the IHCIT UNAH. Precipitation data is recorded in millimeters (mm) and collected daily by the UNAH meteorological station. To achieve the best accuracy in estimation and prediction of the time series, an approach based on neural network models is employed, contrasting the results with statistical models ARIMA. This combination of approaches allows for a thorough exploration of historical and current precipitation data to identify patterns, trends, and future projections. By implementing these models, the aim is to address the inherent complexity in climate prediction. The results obtained in this research will establish the feasibility of implementing this methodology in various meteorological stations to obtain a more precise understanding of precipitation behavior in the Honduran region. These findings may directly inform the formulation of strategies for adaptation and mitigation of climate change in Honduras, representing a valuable contribution to addressing this crucial challenge at a regional level.

RESUMEN. Este estudio aborda el desafío de pronosticar la precipitación en Honduras a través de un análisis de series temporales, utilizando datos proporcionados por el IHCIT UNAH. Los datos de las precipitaciones se registra en milímetros (mm) y se recopilan diariamente por la estación meteorológica de la UNAH. Para lograr la mejor precisión de la estimación y la predicción de la serie temporal, se emplea un enfoque basado en modelos redes neuronales, contrastando los resultados con los modelos estadísticos ARIMA. Esta combinación de enfoques permite una exploración minuciosa de los datos históricos y actuales de precipitación con el fin de identificar patrones, tendencias y proyecciones futuras. Al implementar estos modelos, se busca resolver la complejidad inherente en la predicción climática. Los resultados obtenidos en esta investigación permitirán establecer la factibilidad de implimentar esta metodología en diversas estaciones meteorológicas para obtener una visión más precisa del comportamiento de la precipitación en la región hondureña. Estos hallazgos pueden proporcionar directamente la formulación de estrategias para la adaptación y mitigación del cambio climático en Honduras, lo que representa una contribución valiosa para abordar este desafío crucial a nivel regional.

Fecha: 2 de Marzo de 2024.

Palabras y frases clave. Deep Learnig, Machine Learning, Redes Neuronales, Series Temporales, ARIMA, Cambio Climático, Precipitación.

1. INTRODUCCIÓN

El cambio climático es uno de los desafíos más apremiantes de nuestra era [13], con implicaciones profundas para el medio ambiente, la economía y la sociedad [17]. La capacidad de comprender y predecir con precisión los cambios en la precipitación es esencial para desarrollar estrategias de mitigación y adaptación efectivas. Sin embargo, a pesar de los avances en nuestra comprensión del cambio climático, la predicción precisa de la precipitación sigue siendo un desafío complejo debido a la naturaleza dinámica y multifacética del clima.

En este sentido, los modelos estadísticos se han utilizado ampliamente para analizar datos climáticos y realizar predicciones [1]. Estos modelos, como el análisis de series temporales univariadas, ofrecen un enfoque sólido para comprender la variabilidad de la precipitación y evaluar la incertidumbre en las predicciones [5]. Sin embargo, su capacidad para capturar la complejidad de los datos climáticos puede ser limitada [2]. Es aquí donde las redes neuronales emergen como una herramienta prometedora. Las redes neuronales son capaces de aprender patrones complejos y no lineales en conjuntos de datos, lo que las hace especialmente adecuadas para modelar fenómenos climáticos [2].

Al analizar métodos de redes neuronales con modelos estadísticos, podemos aprovechar lo mejor de ambos enfoques para mejorar nuestra capacidad predictiva, lo que a su vez contribuirá al desarrollo de estrategias más efectivas de adaptación y mitigación del cambio climático en la región. Nos adentramos en una exploración de diversos enfoques para la estimación de la precipitación en Honduras. Comenzaremos con un análisis exhaustivo de la serie temporal antes de aplicar los modelos. Posteriormente utilizando modelos estadísticos como los modelos Autoregresivos (AR), de Medias Móviles (MA), Autoregresivos de Medias Móviles (ARMA). Estos modelos, reconocidos por su eficacia en el análisis de series temporales [4], nos servirán como punto de partida.

Es crucial destacar el proceso de comparativa entre diversos métodos de evaluación de modelos. Inicialmente, se procederá a la selección del mejor modelo AR, MA, ARMA mediante el criterio de Verosimilitud, el cual permitirá determinar el número óptimo de retrasos (también se le suele llamar lags) para cada modelo. Posteriormente, cada uno de estos mejores modelos será sometido a una comparación utilizando tres métodos de evaluación: el criterio de Información de Akaike (AIC), el criterio de Información Bayesiana (BIC) y el criterio de Verosimilitud [20]. Este proceso identificará el modelo con el mejor desempeño entre los cuatro enfoques estadísticos evaluados. Una vez identificado el mejor modelo estadístico, se procederá a compararlo con el modelo de redes neuronales utilizando el error cuadrático medio como medida de evaluación.

Esta evaluación nos proporcionará el mejor modelo de estimación y predicción de la precipitación en Honduras. Además esta investigación presenta múltiples implicaciones y beneficios significativos:

1. *Información para la toma de decisiones:* Proporciona información valiosa que puede guiar la formulación de políticas y estrategias de mitigación y adaptación ante el cambio climático.
2. *Avance científico:* Al Implementar Redes Neuronales, esta investigación ofrece una perspectiva innovadora y más completa del sistema climático, lo que puede

conducir a una comprensión más profunda de sus mecanismos y procesos subyacentes.

3. *Líneas de Investigación*: El cambio climático forma parte de las áreas de investigación de la Universidad Nacional Autónoma de Honduras, evidenciando su compromiso con el abordaje de esta problemática desde un enfoque académico.

2. ANTECEDENTES

Los modelos Estadísticos han sido ampliamente utilizados en el análisis y predicción de series temporales en diversos campos [14], incluido el estudio del cambio climático [12]. Antes del advenimiento de modelos más avanzados y específicos para el análisis climático, como los modelos climáticos acoplados (CMIP) y los modelos de regresión no lineal, los modelos ARIMA proporcionaron una herramienta fundamental para entender las tendencias y variaciones en los datos climáticos. Los antecedentes del modelo ARIMA en el contexto del cambio climático se remontan a las décadas de 1960 y 1970, cuando los investigadores comenzaron a aplicar métodos estadísticos para analizar y pronosticar patrones climáticos [5].

En ese momento, el enfoque principal estaba en comprender la variabilidad climática a corto y mediano plazo, así como en identificar posibles tendencias a largo plazo en variables climáticas como la temperatura, la precipitación y los patrones de presión atmosférica. Uno de los primeros usos significativos de los modelos ARIMA en el estudio del cambio climático fue para analizar y predecir la variabilidad interanual y decadal en fenómenos climáticos importantes, como El Niño y La Niña [1].

Los científicos utilizaron modelos ARIMA para identificar patrones en las series temporales de temperatura del océano y otros parámetros climáticos asociados con estos eventos climáticos extremos [1]. A medida que avanzaba la investigación, los modelos ARIMA se aplicaron a una variedad de variables climáticas en diferentes escalas espaciales y temporales. Se utilizaron para analizar la variabilidad estacional, las tendencias a largo plazo y los ciclos climáticos en regiones específicas, así como para realizar pronósticos climáticos a corto y mediano plazo. Sin embargo, a medida que aumentaba la complejidad de los modelos climáticos y se disponía de datos más detallados y extensos, los modelos ARIMA comenzaron a complementarse con enfoques más sofisticados, como los modelos de espacio-estado y los modelos de series temporales no lineales.

Aún así, los modelos ARIMA siguen siendo útiles en muchas aplicaciones climáticas, especialmente en el análisis de series temporales cortas y en la detección de cambios y anomalías en los datos climáticos históricos. Durante las últimas décadas, las redes neuronales han emergido como herramientas poderosas en diversas disciplinas científicas [2], incluida la climatología, debido a su capacidad para modelar relaciones complejas y no lineales en conjuntos de datos. Uno de los primeros usos significativos de las redes neuronales en la climatología fue en la predicción de fenómenos climáticos extremos y la modelización de patrones climáticos regionales. En un estudio pionero realizado en 1995, Hsieh et al. aplicaron redes neuronales para predecir la intensidad de los huracanes en el Atlántico Norte, demostrando su eficacia en la predicción de eventos meteorológicos críticos [7].

Las redes neuronales se utilizaron en una variedad de aplicaciones climáticas [2], incluida la modelización del cambio climático global y la predicción de variables climáticas clave, como la temperatura y la precipitación. En un estudio relevante llevado a cabo en 2008, Wang et al. utilizaron redes neuronales para mejorar la precisión de las proyecciones del clima futuro, destacando su capacidad para capturar patrones climáticos complejos y no lineales [19]. Además de la predicción del clima, las redes neuronales también se han empleado en la detección de señales climáticas, el análisis de datos climáticos y la evaluación de la vulnerabilidad ante el cambio climático.

En un trabajo significativo publicado en 2013, Liang et al. utilizaron redes neuronales para identificar patrones de cambio climático en series temporales de datos climáticos observados, contribuyendo así a una comprensión más profunda de las tendencias climáticas globales [10].

3. ANÁLISIS PROSPECTIVO AL CAMBIO CLIMÁTICO

3.1. Series de Tiempo. En esta sección exploraremos los conceptos clave para el análisis de series temporales que serán fundamentales para nuestra investigación. Destacaremos la importancia de la estacionariedad, así como las funciones de autocovarianza y autocovarianza muestral, que nos serán de gran utilidad para analizar y comprender los datos temporales que estudiaremos.

Definición 1. Una serie temporal se define como una sucesión de variables aleatorias indexadas por el tiempo. Sea $\{X_t : t \in T\}$ una colección de variables aleatorias, donde T representa el conjunto de índices temporales [14].

3.1.1. Estacionariedad. La Estacionariedad se refiere a patrones repetitivos o cíclicos que ocurren en una serie temporal con una periodicidad específica [5]. La Estacionariedad puede clasificarse en débil o estricta. La Estacionariedad estricta es más difícil de probar empíricamente, pero la Estacionariedad débil puede ser equivalente a la Estacionariedad estricta si la serie temporal sigue una distribución normal [12]. En esta investigación, estamos principalmente preocupado por series débilmente estacionarias.

Definición 2. Una serie temporal $\{X_t, t \in \mathbb{Z}\}$ se considera débilmente estacionaria si su media, varianza y covarianza son constantes, en la condición de estacionariedad débil, asumimos que los primeros dos momentos de X_t son finitos.

3.1.2. Función de Correlación y Autocorrelación. Para evaluar el grado de dependencia en los datos (Lags) y seleccionar un modelo que refleje esto, utilizamos es la función de *autocorrelación* (ACF) de los datos y *autocorrelación Parcial* (PACF). Mientras que la ACF muestra la correlación entre una serie de tiempo y sus lags anteriores, la PACF identifica relaciones directas entre la serie de tiempo y sus lags individuales. [12]

Definición 3. Función de Autocorrelación: Consideremos una serie temporal débilmente estacionaria $\{X_t, t \in \mathbb{Z}\}$. Cuando la dependencia lineal entre X_t y sus valores pasados X_{t-i} es de interés, el concepto de correlación se generaliza a la autocorrelación

Definición 4. El coeficiente de correlación entre X_t y $X_{t-\tau}$ se llama autocorrelación con desfase (lags) de τ de X_t y comúnmente se denota por ρ_τ , que bajo la suposición de estacionariedad débil es una función de τ únicamente. Específicamente, definimos:

$$(3.1) \quad \rho_\tau = \frac{\text{Cov}(X_t, X_{t-\tau})}{\sigma_\tau \sigma_{t-\tau}}$$

3.1.3. Ruido Blanco. El ruido blanco es una señal aleatoria que exhibe fluctuaciones aleatorias sin ninguna estructura o patrón discernible. Es una herramienta esencial en el análisis de series temporales, ya que su objetivo principal es modelar la variabilidad aleatoria inherente en los datos. Además, ayuda a distinguir entre la señal (patrones de interés) y el ruido (fluctuaciones aleatorias) en un conjunto de datos observados, lo que facilita el análisis y la interpretación de las tendencias y patrones presentes en los datos [12].

Definición 5. Una serie temporal X_t se llama ruido blanco si $\{X_t\}$ es una secuencia de variables aleatorias independientes e idénticamente distribuidas con media y varianzas finitas.

3.1.4. Criterios de Comparación. Existen varios criterios de Comparación para determinar el orden p de un modelo. Todos ellos se basan en la verosimilitud [20]. El criterio de información de Akaike (AIC) se define como

$$\text{AIC} = -\frac{2}{T} \ln(\text{verosimilitud}) + \frac{2}{T} \times (\text{número de parámetros}),$$

donde la función de verosimilitud se evalúa en las estimaciones de máxima verosimilitud y T es el tamaño de la muestra. El primer término del AIC mide la bondad de ajuste del modelo a los datos, mientras que el segundo término se llama función de penalización del criterio porque penaliza un modelo candidato por el número de parámetros utilizados [14]. Diferentes funciones de penalización resultan en diferentes criterios de información. Otro criterio de función comúnmente utilizado es el criterio de información Schwarz-Bayesiano (BIC) [20]

$$\text{BIC}(P) = \ln(\hat{\sigma}_P^2) + \ln(T) \frac{P}{T}.$$

La penalización por cada parámetro utilizado es de 2 para AIC y $\ln(T)$ para BIC. Por lo tanto, en comparación con AIC, BIC tiende a seleccionar un modelo más bajo cuando el tamaño de la muestra es moderado o grande.

3.1.5. Regla de Selección. Para utilizar AIC para seleccionar un modelo en la práctica, se calcula $\text{AIC}(p)$ para $p = 0, \dots, P$, donde P es un entero positivo predefinido, y se selecciona el orden k que tenga el valor mínimo de AIC. La misma regla se aplica a BIC. No hay evidencia que sugiera que un enfoque sea mejor que el otro en una aplicación real [20]. La información sustantiva del problema en estudio y la simplicidad son dos factores que también juegan un papel importante en la elección de un modelo para una serie de tiempo dada.

3.1.6. Comprobación del Modelo. Un modelo ajustado debe examinarse cuidadosamente para verificar posibles deficiencias. Si el modelo es adecuado, entonces la serie de residuos debería comportarse como un ruido blanco. El ACF de los residuos pueden usarse para verificar la cercanía de \hat{a}_t a un ruido blanco [11].

3.2. Modelo AR. Definimos un modelo como autorregresivo si la variable endógena de un período t es explicada por las observaciones de ella misma correspondientes a períodos anteriores añadiéndose, como en los modelos estructurales, un término de error [4].

Definición 6. Sea r_t una serie de tiempo, el modelo AR de orden 1 es define como

$$(3.2) \quad r_t = \phi_0 + \phi_1 r_{t-1} + a_t$$

donde $\{a_t\}$ se asume ser una serie de ruido blanco con media cero y varianza σ_a^2 y $\phi_0 + \phi_1 r_{t-1}$ se conoce como el polinomio de retraso [14].

Este modelo está en la misma forma que el conocido modelo de regresión lineal simple en el cual r_t es la variable dependiente y r_{t-1} es la variable explicativa [5]. Una generalización directa del modelo AR(1) es el modelo AR(p):

$$(3.3) \quad r_t = \phi_0 + \phi_1 r_{t-1} + \dots + \phi_p r_{t-p} + a_t$$

donde p es un entero no negativo y $\{a_t\}$ está definido en la Ecuación (3.2). Este modelo indica que las p variables pasadas r_{t-i} (para $i = 1, \dots, p$) determinan conjuntamente la expectativa condicional de r_t dado los datos pasados.

3.2.1. Identificación del Orden. En una serie temporal estacionaria la Función de Autocorrelación Parcial (PACF) es una función útil para determinar el orden p de un modelo AR. Una forma simple pero efectiva de introducir la PACF es considerar los siguientes modelos AR en órdenes consecutivos:

$$\begin{aligned} r_t &= \phi_{0,1} + \phi_{1,1} r_{t-1} + e_{1,t}, \\ r_t &= \phi_{0,2} + \phi_{1,2} r_{t-1} + \phi_{2,2} r_{t-2} + e_{2,t}, \\ &\vdots \end{aligned}$$

donde $\phi_{0,j}$, $\phi_{i,j}$, y $\{e_{j,t}\}$ son, respectivamente, el término constante, el coeficiente de r_{t-i} y el término de error de un modelo AR(j). Estos modelos están en forma de una regresión lineal múltiple y pueden ser estimados por el método de mínimos cuadrados. La estimación $\hat{\phi}_{1,1}$ de la primera ecuación se llama la PACF de muestra de lag-1 de r_t . La estimación $\hat{\phi}_{2,2}$ de la segunda ecuación es la PACF de muestra de lag-2 de r_t , y así sucesivamente [12].

3.3. Modelo MA. Hay varias formas de introducir modelos MA. Un enfoque es tratar el modelo como una simple extensión de series de ruido blanco. Otro enfoque es tratar el modelo como un modelo AR de orden infinito con algunas restricciones de parámetros [14]. Adoptamos el segundo enfoque. No hay una razón particular, excepto la simplicidad, para asumir a priori que el orden de un modelo AR es finito. Podemos considerar, al menos en teoría, un modelo AR con orden infinito como

$$(3.4) \quad r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + \dots + a_t.$$

Sin embargo, dicho modelo AR no es realista porque tiene un número infinito de parámetros. Una forma de hacer que el modelo sea práctico es asumir que los coeficientes ϕ_i satisfacen algunas restricciones para que estén determinados por un número finito de parámetros. Un caso especial de esta idea es

$$(3.5) \quad r_t = \phi_0 - \theta_1 r_{t-1} - \theta_2 r_{t-2} - \theta_3 r_{t-3} - \dots + a_t,$$

donde los coeficientes dependen de un solo parámetro θ_1 a través de $\phi_i = -\theta_1^i$ para $i \geq 1$. Para que el modelo en la Ecuación (3.5) sea estacionario, θ_1 debe ser

menor que 1 en valor absoluto; de lo contrario, θ_i^1 y la serie explotarán. Dado que $|\theta_1| < 1$, tenemos $\theta_i^1 \rightarrow 0$ a medida que $i \rightarrow \infty$. Por lo tanto, la contribución de r_{t-i} a r_t decae exponencialmente a medida que i aumenta. Esto es razonable ya que la dependencia de una serie estacionaria r_t en su valor rezagado r_{t-i} , si la hay, debería decaer con el tiempo [12].

Definición 7. La forma general de un modelo MA(1) es

$$(3.6) \quad r_t = c_0 + a_t - \theta_1 a_{t-1}$$

donde c_0 es una constante y $\{a_t\}$ es una serie de ruido blanco.

un modelo MA(q) es de la forma

$$(3.7) \quad r_t = c_0 + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

donde $q > 0$ [14].

3.3.1. Identificación del Orden. El ACF es útil para identificar el orden de un modelo MA [14]. Para una serie temporal r_t con ACF ρ_l , si $\rho_l \neq 0$, pero $\rho_l = 0$ para $l > q$, entonces r_t sigue un modelo MA(q).

3.4. Modelo ARMA. En algunas aplicaciones, los modelos AR o MA discutidos en las secciones anteriores se vuelven engorrosos porque puede ser necesario un modelo de alto orden con muchos parámetros para describir adecuadamente la estructura dinámica de los datos. Para superar esta dificultad, se introducen los modelos autorregresivos de medias móviles (ARMA) [15]. Básicamente, un modelo ARMA combina las ideas de los modelos AR y MA en una forma compacta de manera que el número de parámetros utilizados se mantenga pequeño, logrando la parsimonia en la parametrización. En esta sección, estudiamos el modelo ARMA(1,1) más simple.

Definición 8. Una serie temporal r_t sigue un modelo ARMA(1,1) si satisface la siguiente ecuación:

$$(3.8) \quad r_t - \phi_1 r_{t-1} = \phi_0 + a_t - \theta_1 a_{t-1}$$

donde $\{a_t\}$ es una serie de ruido blanco. El lado izquierdo de la ecuación (3.8) es el componente AR del modelo y el lado derecho da el componente MA. El término constante es ϕ_0 . Para que este modelo sea significativo, necesitamos $\phi_1 = \theta_1$; de lo contrario, habría una cancelación en la ecuación y el proceso se reduciría a una serie de ruido blanco [14].

Definición 9. Un modelo ARMA(p,q) general tiene la forma:

$$(3.9) \quad r_t = \phi_0 + \sum_{i=1}^p \phi_i r_{t-i} + a_t - \sum_{i=1}^q \theta_i a_{t-i},$$

donde $\{a_t\}$ es una serie de ruido blanco y p y q son enteros no negativos. Los modelos AR y MA son casos especiales del modelo ARMA(p,q).

3.4.1. Identificación del Orden. El ACF y el PACF no son informativos para determinar el orden de un modelo ARMA [4]. La función que nos ayudará a identificar el orden es el "Likelihood Ratio Test" [3]. El Likelihood Ratio Test (LRT) es una prueba estadística utilizada para comparar la adecuación de dos modelos estadísticos, evaluando si un modelo más complejo es significativamente mejor que un modelo más simple al comparar la verosimilitud de los datos entre ambos modelos..

3.4.2. *Series temporales estacionarias de tendencia.* Un modelo estrechamente relacionado que exhibe una tendencia lineal es el modelo de series temporales estacionarias de tendencia [12].

Definición 10. Sea r_t una serie temporal estacionaria, un modelo de series temporales estacionarias de tendencia

$$(3.10) \quad p_t = \beta_0 + \beta_1 t + r_t,$$

donde p_t crece linealmente en el tiempo con una tasa β_1 .

Definición 11. La serie estacionaria de tendencia puede transformarse en una estacionaria eliminando la tendencia temporal mediante un análisis de regresión lineal simple.

3.4.3. *Modelos Generales No Estacionarios de Raíz Unitaria.* Considera un modelo ARMA. Si uno extiende el modelo permitiendo que el polinomio AR tenga 1 como raíz característica, entonces el modelo se convierte en el conocido modelo autoregresivo integrado de media móvil (ARIMA). Se dice que un modelo ARIMA es no estacionario debido a raíz unitaria porque su polinomio AR tiene una raíz unitaria. Un enfoque convencional para manejar la no estacionariedad debido a la raíz unitaria es utilizar la diferenciación.

3.4.4. *Diferenciación.* Algunas series temporales no son estacionarias debido a tendencias o efectos estacionales [5]. Las series no estacionarias pueden transformarse en series estacionarias mediante la diferenciación. Una vez diferenciadas, podemos ajustar un proceso ARMA en ellas. Estos procesos son conocidos como procesos autoregresivos integrados de media móvil (o ARIMA), ya que la serie diferenciada necesita ser sumada o integrada para recuperar la serie original [21].

Definición 12. Una serie temporal y_t se dice que es un proceso ARIMA(p, 1, q) si la serie de cambios $c_t = y_t - y_{t-1} = (1 - B)y_t$ sigue un modelo ARMA(p,q) estacionario. Donde $c_t = y_t - y_{t-1}$ se refiere como la primera serie diferenciada de y_t .

Definición 13. Una serie temporal y_t puede contener múltiples raíces unitarias y necesita ser diferenciada varias veces para volverse estacionaria. Además, si s_t sigue un modelo ARMA(p,q), entonces y_t es un proceso ARIMA(p, 2, q) [21].

3.4.5. *Prueba de Raíz Unitaria.* Para probar si una serie temporal p_t es estacionaria, empleamos los modelos

$$(3.11) \quad p_t = \phi_1 p_{t-1} + e_t$$

$$(3.12) \quad p_t = \phi_0 + \phi_1 p_{t-1} + e_t$$

donde e_t denota el término de error, y consideramos la hipótesis nula $H_0 : \phi_1 = 1$ frente a la hipótesis alternativa $H_a : \phi_1 < 1$. Este es el conocido problema de prueba de raíz unitaria; [3]. Una estadística de prueba conveniente es la razón t de la estimación de mínimos cuadrados (LS) de ϕ_1 bajo la hipótesis nula. Para la Ecuación (3.11), el método de mínimos cuadrados (LS) proporciona

$$(3.13) \quad \hat{\phi}_1 = \frac{\sum_{t=1}^T p_{t-1} p_t}{\sum_{t=1}^T p_{t-1}^2}, \quad \hat{\sigma}_e^2 = \frac{\sum_{t=1}^T (p_t - \hat{\phi}_1 p_{t-1})^2}{T - 1},$$

donde $p_0 = 0$ y T es el tamaño de la muestra. La razón t es

$$(3.14) \quad DF \equiv t \text{ ratio} = \frac{\hat{\phi}_1 - 1}{\text{std}(\hat{\phi}_1)} = \frac{\sum_{t=1}^T p_{t-1} e_t}{\hat{\sigma}_e \sqrt{\frac{T}{\sum_{t=1}^T p_{t-1}^2}}},$$

que comúnmente se conoce como la prueba de Dickey-Fuller (DF). Si $\{e_t\}$ es una serie de ruido blanco con momentos finitos de orden ligeramente mayor que 2, entonces la estadística DF converge a una función de la movilidad Browniana estándar a medida que $T \rightarrow \infty$ [16].

3.4.6. Series Temporales Estacionales. En algunas aplicaciones, la estacionalidad es de importancia secundaria y se elimina de los datos, lo que resulta en una serie temporal ajustada estacionalmente que luego se utiliza para hacer inferencias [12].

Definición 14. El procedimiento para eliminar la estacionalidad de una serie temporal se denomina ajuste estacional [12].

En otras aplicaciones como la predicción, la estacionalidad es tan importante como otras características de los datos y debe manejarse en consecuencia. Dado que la predicción es un objetivo importante para esta investigación, nos centramos y discutimos algunos modelos que son útiles en el modelado de series temporales estacionales.

3.5. Inteligencia Artificial. Todos los conceptos presentados en esta sección se tomaron de [18]. La inteligencia artificial (IA) abarca varias disciplinas interrelacionadas, entre las que se incluyen el aprendizaje automático (Machine Learning: ML), el aprendizaje profundo (Deep Learning) y la minería de datos (Data Mining). Dentro de la IA, dos subconjuntos principales son el aprendizaje automático y el aprendizaje profundo. El aprendizaje automático, también conocido como ML, se centra en desarrollar algoritmos que pueden aprender de los datos sin ser programados explícitamente para tareas específicas. Por otro lado, el aprendizaje profundo se enfoca en algoritmos inspirados en la estructura y funcionamiento del cerebro humano. En esta investigación se centrará en esta última.

3.5.1. Redes Neuronales. El cerebro humano consta de aproximadamente 10 mil millones de neuronas, cada una de ellas establece conexiones con alrededor de 12 mil neuronas adicionales [9]. Estas neuronas reciben señales electroquímicas de otras neuronas a través de sus dendritas. Si la suma de estas señales es lo suficientemente fuerte como para activar la neurona, esta transmite una señal electroquímica a lo largo de su axón.

Los sistemas neuronales artificiales (ANS) se inspiran en la estructura del sistema nervioso biológico para desarrollar sistemas de procesamiento paralelos, distribuidos y adaptativos, que pueden exhibir cierto grado de comportamiento *inteligente*. Las neuronas artificiales se organizan en capas que forman una red neuronal. Estas redes neuronales, ya sea individualmente o interconectadas, pueden constituir sistemas de procesamiento global. La siguiente figura presenta un modelo básico de una neurona artificial y su correspondencia con la neurona biológica. Este modelo consta de los siguientes elementos:

- Un conjunto de pesos sinápticos w_i asociados con cada una de las entradas $x(t)$.

- Una regla de propagación, típicamente expresada como la suma ponderada de las entradas: $h_i(t) = \sum w_{ij}x_j(t)$.
- Una función de activación f_i para calcular la salida de la neurona: $y_i(t) = f_i(h_i(t))$.

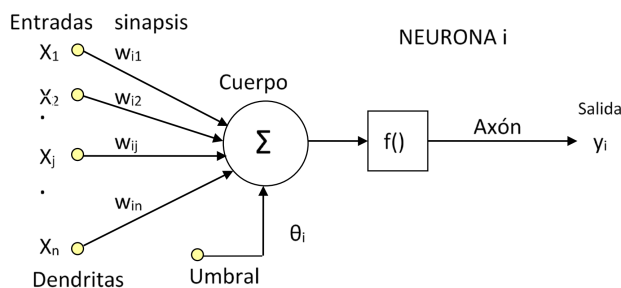


FIGURA 1. Modelo Estándar

Definición 15. Una red neuronal artificial (RNA), también conocida como red neuronal está formada por la interconexión de muchas unidades de procesamiento llamadas neuronas, formadas por bloques no lineales distribuidas en toda la red neuronal.

En el contexto de las redes neuronales artificiales, las neuronas suelen agruparse en capas, y una o más capas constituyen la red neuronal en su totalidad. En las redes neuronales, se distinguen tres tipos principales de capas: de entrada, oculta y de salida.

La capa de entrada, está compuesta por neuronas que reciben datos. Por otro lado, la capa de salida contiene neuronas que proporcionan la salida o respuesta final de la red neuronal. Por último, la capa oculta o intermedia no tiene conexiones directas con el entorno externo y es donde se lleva a cabo el procesamiento interno de la información. Esta capa actúa como un puente entre la capa de entrada y la de salida, extrayendo características relevantes de los datos de entrada y generando representaciones útiles para la tarea en cuestión.

3.5.2. Función de Activación. Una función de activación, que simultáneamente representa la salida de la neurona y su estado de activación, se denota como

$y_i = f_i(h_i) = f_i\left(\sum_{j=0}^n w_{ij}x_j\right)$. Existen varios ejemplos de funciones de activación, entre ellos:

- *Neuronas todo-nada:* También conocidas como dispositivos de umbral, estas neuronas tienen una función escalonada. Para este tipo de función de activación, la salida es 1 si $\sum_{j=1}^n w_{ij}x_j$ es mayor o igual que el umbral θ_i , y 0 en caso contrario.

- *Neurona continua sigmoidea:* Para obtener una salida continua, es común utilizar una función sigmoidea. la función más común y la que se utilizará es

$$y_i = \frac{1}{1 + e^{-\left(\sum_{j=1}^n w_{ij}x_j - \theta_i\right)}}, \text{ donde } y_i \text{ pertenece al intervalo } [0, 1].$$

Usaremos la combinación de estas funciones de activación para crear la estructura de la red neuronal.

3.5.3. *Capas.* Para representar adecuadamente las múltiples entradas de una neurona, se emplea notación matricial para los pesos W . Cada entrada individual p_1, p_2, \dots, p_R está asociada con pesos específicos $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ en la matriz de pesos W . La salida n puede expresarse mediante la siguiente ecuación:

$$(3.15) \quad n = w_{1,1} \cdot p_1 + w_{1,2} \cdot p_2 + \dots + w_{1,R} \cdot p_R + b$$

Las matrices de peso conectadas a las entradas se denominan matrices de ponderación o matriz de pesos de entrada, mientras que aquellas que utilizan la salida proveniente de otra capa se denominan matrices de pesos. Los superíndices identifican la fuente (segundo índice) y el destino (primer índice) para distinguir los pesos y otros componentes de la red. En la Figura 2 se presenta una red neuronal de entrada múltiple compuesta por una capa. Aquí, S representa la cantidad de neuronas de la capa 1, y R denota el número de elementos del vector de entrada. Se observa que para la matriz de ponderación o pesos conectada al vector de entrada p , se le asigna $I_{1,1}$, donde el segundo índice indica la fuente 1 y el primer índice también es 1. Los sesgos, las entradas netas y las salidas de la capa 1 tienen un superíndice 1, indicando que pertenecen a la primera capa.

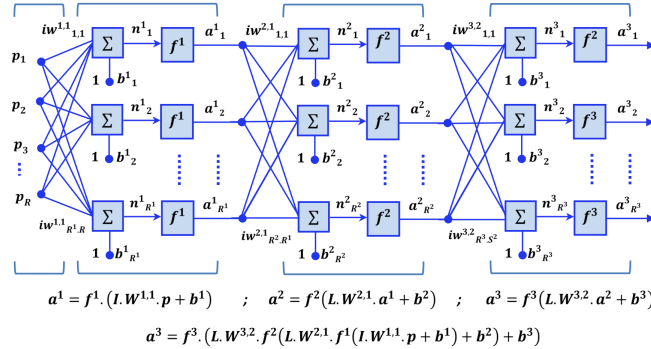


FIGURA 2. Red Neuronal Multicapa

3.5.4. *Clasificación de Redes Neuronales.* Las redes neuronales pueden clasificarse según su aplicación o uso:

- *Clasificación:* En este caso, las redes neuronales se utilizan como clasificadores, donde se asigna una clase discreta a un vector de entradas.
- *Regresión:* Las redes neuronales también pueden actuar como regresores, asignando un vector de salida continuo o analógico a un vector de entrada continuo.

Clasificación según su topología o arquitectura

- Red neuronal Monocapa – Perceptrón simple: Es el modelo más simple de red neuronal artificial.
- Red neuronal Multicapa: Consta de múltiples capas de neuronas interconectadas.

La clasificación de las redes neuronales se puede abordar desde distintos ángulos:

- Supervisado: Durante el entrenamiento, la red recibe patrones de entrada y las salidas deseadas asociadas. Ajusta sus parámetros internos en base a estos datos para minimizar la discrepancia entre las salidas predichas y las deseadas.
- No supervisado: En este caso, la red no dispone de salidas deseadas durante el entrenamiento. Se ajusta en función de patrones y similitudes encontradas en los datos de entrada, utilizando técnicas como el agrupamiento o el análisis de componentes principales (PCA).
- Reforzado: La red aprende a través de la retroalimentación proporcionada por un entorno externo, donde se evalúa si las respuestas son correctas o incorrectas.

Estos modelos se construyen utilizando ecuaciones diferenciales o en diferencia para minimizar errores en las salidas. Esta investigación se trabajará como un problema de Regresión supervisado con una Red Neuronal Multicapa.

3.5.5. Algoritmos de Entrenamiento de Redes: ADAM. Los conceptos presentados fueron extraídos de [8]. ADAM (Adaptive Moment Estimation) es un método de optimización popularmente utilizado en el entrenamiento de redes neuronales. Fue propuesto por Diederik P. Kingma y Jimmy Ba en 2014. ADAM combina conceptos de dos técnicas de optimización previas: RMSProp (Root Mean Square Propagation) y Momentum.

- *Momentum*: ADAM utiliza el concepto de momento para ayudar a acelerar el proceso de optimización. El momento se calcula como una combinación lineal entre el gradiente actual y el momento anterior, lo que ayuda a mantener una dirección coherente en la actualización de los pesos.
- *RMSProp*: ADAM también integra la idea de RMSProp, que adapta las tasas de aprendizaje de cada parámetro de la red de manera individual. Esto se hace mediante el cálculo de un promedio exponencial ponderado de los cuadrados de los gradientes anteriores, lo que permite ajustar la velocidad de aprendizaje de cada parámetro de manera adaptativa.

En resumen, ADAM es un método de optimización eficaz para entrenar redes neuronales, ya que combina las ventajas del momento y RMSProp para adaptar de manera dinámica las tasas de aprendizaje de los parámetros de la red, lo que conduce a un entrenamiento más rápido y eficiente.

3.6. Descripción de Experimentos. Nuestro estudio se sumerge en una exploración detallada de múltiples enfoques para estimar la precipitación en Honduras. Comenzamos este proceso con un análisis minucioso de la serie temporal de datos antes de aplicar cualquier modelo [12].

Los datos utilizados en este estudio fueron proporcionados por el Instituto Hondureño de Ciencias de la Tierra (IHCIT) y consiste en registros diarios de precipitación desde 1979 hasta 2023. Durante este proceso, se realizó una limpieza de datos para

manejar valores nulos y la presencia de cadenas de caracteres “TZR” (precipitaciones pequeñas). Los valores nulos fueron rellenados utilizando la técnica de relleno hacia adelante, mientras que las cadenas de caracteres fueron convertidas a valores numéricos apropiados. Además, se ajustó la frecuencia de los datos a diaria y se realizaron análisis de residuos, prueba de estacionariedad [3], funciones de autocorrelación (ACF) y autocorrelación parcial (PACF) para evaluar la estacionalidad y la autocorrelación en los datos.

Posteriormente, se dividió los datos en conjuntos de entrenamiento y prueba, donde el 80 % de los datos se destinaron al conjunto de entrenamiento y el 20 % restante al conjunto de prueba. Después de la preparación de los datos, se procedió a la selección de los mejores modelos AR, MA, ARMA. Utilizamos los criterios AIC y BIC [20] para determinar los lags óptimos, es decir, para encontrar el modelo más simple que mejor se ajuste a los datos. En casos donde los retrasos (lags) difieren entre los criterios, aplicamos el “Likelihood Ratio Test” [3] para evaluar si un modelo con un número mayor de lags, es decir, una mayor complejidad, mejora significativamente la precisión del modelo en comparación con uno más simple. Una vez seleccionado el mejor modelo estadístico entre los modelos AR, MA, ARMA, procedimos a evaluar su capacidad predictiva. Realizamos una predicción de 61 días y comparamos los resultados con los datos de prueba.

Este proceso meticuloso nos permitió identificar el modelo con el mejor rendimiento entre los enfoques estadísticos analizados. Posteriormente, comparamos el modelo estadístico seleccionado con el modelo basado en redes neuronales multicapa utilizando el error cuadrático medio como medida de evaluación para ambos modelos. Esta comparación proporcionó una evaluación completa de la capacidad predictiva de cada modelo, ayudándonos a determinar el más adecuado para estimar la precipitación en Honduras.

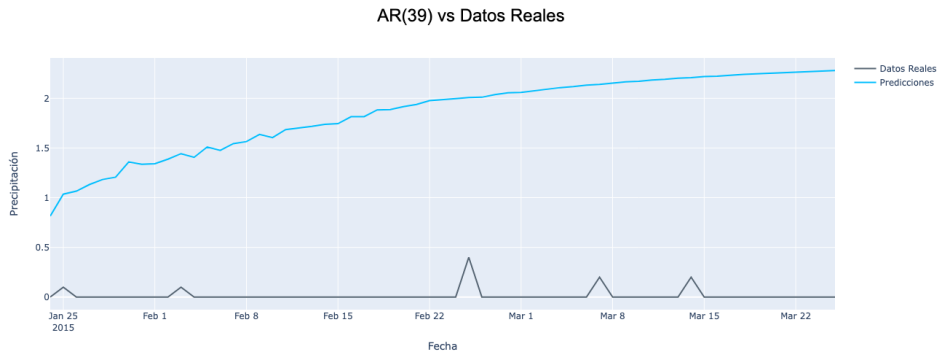
3.7. Experimentos.

3.7.1. Preprocesamiento de Datos. Tras el preprocesamiento de los datos, se llevó a cabo un análisis exhaustivo de la serie temporal. Se realizó una visualización de la serie de tiempo para identificar tendencias y patrones, seguido de la aplicación de la prueba de Dickey-Fuller aumentada (ADF) para confirmar la estacionariedad de la serie [3]. Los resultados de la prueba de ADF revelaron que la serie de tiempo es estacionaria, con una estadística ADF de -15.3976, que se sitúa por debajo del umbral crítico del 5 % de ADF (-2.86176). Además, se observó un número de lag de 24 y un valor p extremadamente bajo de $3.248e^{-28}$. Estos hallazgos confirman la estacionariedad de la serie de tiempo y sientan las bases para la selección del mejor modelo entre AR, MA, ARMA.

3.7.2. MEJOR MODELO AR. El proceso de determinar el número óptimo de rezagos se realiza considerando múltiples perspectivas. Inicialmente, evaluamos los criterios de comparación de la Función de Autocorrelación Parcial (PACF) y los criterios de Información de Akaike (AIC) y Bayesiano (BIC).

El gráfico de PACF sugiere un número de rezagos de 30, indicando que la precipitación actual depende directamente de los 30 días anteriores. Sin embargo, al analizar los criterios de AIC y BIC, obtenemos números óptimos de rezagos de 34 y 39, respectivamente. Ante esta discrepancia, aplicamos el “Likelihood Ratio Test” [3] para evaluar si el modelo más complejo con 39 rezagos se ajusta significativamente

mejor que el modelo con 34 rezagos. Los resultados confirman que el modelo más complejo es estadísticamente superior, lo que indica que un mayor número de rezagos es preferible para capturar la estructura subyacente de los datos. Por lo tanto, concluimos que el modelo autorregresivo óptimo es AR(39).

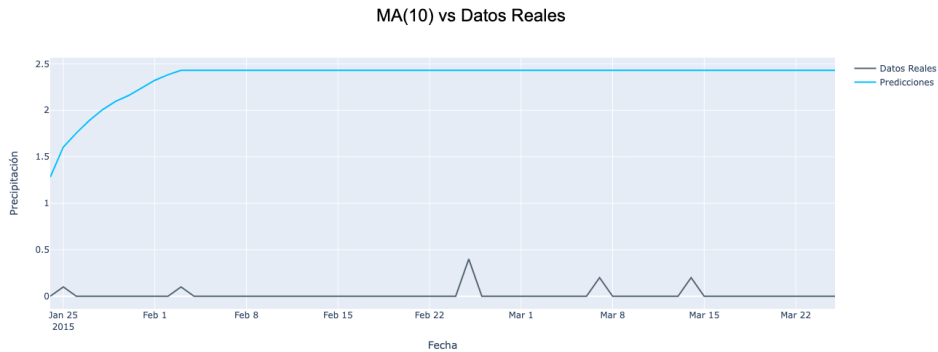


3.7.3. MEJOR MODELO MA. Para determinar el número óptimo de rezagos en el modelo de medias móviles (MA), se evaluaron varios criterios, incluida la Función de Autocorrelación (ACF)[20] y los criterios de información de Akaike (AIC) y Bayesiano (BIC). La ACF sugiere un número de rezagos de 5, mientras que los criterios AIC y BIC indican 10 y 9 rezagos, respectivamente. Ante esta discrepancia, se aplicó el “Likelihood Ratio Test” para evaluar si el modelo más complejo con 10 rezagos es significativamente mejor que el modelo con 5 rezagos.

Los resultados confirman que el modelo con 10 rezagos se ajusta significativamente mejor, lo que indica que un mayor número de rezagos es preferible para capturar la variabilidad en los datos. Por lo tanto, se concluye que el modelo de medias móviles óptimo es MA(10). Es importante destacar que se tomó un máximo de 10 rezagos debido a la complejidad computacional, ya que considerar más rezagos, como 40 (en el caso de AR), resultaría en un aumento significativo en el tiempo de procesamiento.

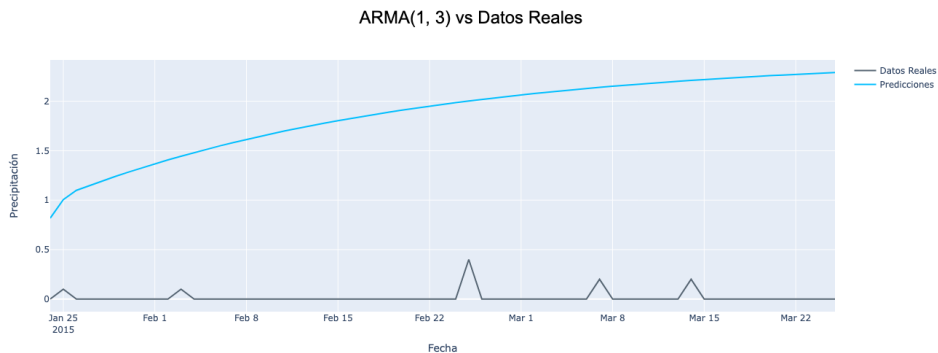
Después de seleccionar el modelo MA(10) como el óptimo, se realizó una comparación adicional entre el modelo MA(10) y un modelo más complejo, MA(30), utilizando el “Likelihood Ratio Test” [3]. Este paso se llevó a cabo para determinar si añadir más rezagos mejoraría significativamente el ajuste del modelo. Los resultados indican que si bien el modelo más complejo muestra un mejor ajuste, esta mejora no es significativa. En otras palabras, aunque el modelo MA(30) captura ligeramente más la variabilidad en los datos, la diferencia en el ajuste entre el modelo MA(10) y MA(30) no es lo suficientemente grande como para justificar la inclusión de 20 rezagos adicionales. Teniendo en cuenta este análisis, se ratifica que el modelo MA(10) es la mejor opción, ya que proporciona un equilibrio entre la capacidad de capturar la complejidad de los datos y la simplicidad del modelo.

3.7.4. MEJOR MODELO ARMA. Para seleccionar el mejor modelo ARMA, nos enfocamos exclusivamente en los criterios de comparación AIC y BIC. Dado que el análisis de la Función de Autocorrelación (ACF) y la Función de Autocorrelación Parcial (PACF) no proporcionaron una guía clara en modelos anteriores. Limitamos



el número máximo de lags a 5 para evitar la complejidad computacional. Según el criterio AIC, el mejor modelo es ARMA(1, 3), mientras que el criterio BIC sugiere ARMA(5, 5) como el más adecuado.

Esta discrepancia en el número de rezagos nos lleva a aplicar el “Likelihood Ratio Test” [3] para evaluar si la mayor complejidad mejora significativamente el modelo. Los resultados revelan que el modelo más simple es estadísticamente superior. Por lo tanto, concluimos que el ARMA óptimo es ARMA(1, 3).



3.7.5. Redes Neuronales Multicapa. Para construir el modelo de redes neuronales, emplearemos el paquete Keras [6]. Consideraremos un número de 365 lags, ya que estamos trabajando con datos diarios y disponemos de 12760 datos de entrenamiento, lo cual proporciona un conjunto de datos lo suficientemente grande para capturar una amplia variedad de patrones temporales. Vale la pena destacar que probamos inicialmente con un número de 30 lags, pero el modelo no se ajustó de manera óptima. Se exploraron diversos métodos de optimización, y encontramos que el método *ADAM* [8] produjo los mejores resultados.

Además, experimentamos con diferentes configuraciones de capas. Observamos que la combinación más efectiva fue una primera capa con 100 neuronas seguida de una segunda capa con 10 neuronas. Incrementar el número de capas no proporcionó mejoras significativas en el rendimiento del modelo; de hecho, una mayor complejidad

con más capas no resultó óptima. Por lo tanto, concluimos que dos capas ocultas ofrecieron los mejores resultados en términos de precisión y eficiencia.

3.8. Comparación de Errores de los Modelos Estadísticos. Para evaluar la eficacia de los modelos, se compararon los errores utilizando diferentes métricas de evaluación. Se calcularon los errores cuadráticos medios (MSE) para cada modelo, los cuales representan la diferencia cuadrada promedio entre los valores reales de precipitación y las predicciones realizadas por los modelos.

Además del MSE, se analizaron otros indicadores de desempeño, como el error absoluto medio (MAE) y el coeficiente de determinación (R^2). El MAE proporciona una medida promedio de la magnitud de los errores de predicción, mientras que el coeficiente de determinación (R^2) indica la proporción de la variabilidad en los datos de precipitación que es explicada por el modelo [20].

Fecha	Obs.	AR		MA		ARMA	
		Pred.	MSE	Pred.	MSE	Pred.	MSE
2015-01-24	0.0	0.8155	3.529	1.2818	5.5274	0.8181	3.5608
2015-01-25	0.1	1.0355	3.529	1.6048	5.5274	1.0051	3.5608
2015-01-26	0.0	1.0658	3.529	1.7561	5.5274	1.0995	3.5608
2015-01-27	0.0	1.1322	3.529	1.8913	5.5274	1.1484	3.5608
2015-01-28	0.0	1.1829	3.5299	2.0070	5.5274	1.1955	3.5608
...
2015-03-22	0.0	2.2650	3.529	2.4301	5.5274	2.2718	3.5608
2015-03-23	0.0	2.2691	3.529	2.4301	5.5274	2.2782	3.5608
2015-03-24	0.0	2.2753	3.529	2.4301	5.5274	2.2844	3.5608
2015-03-25	0.0	2.2795	3.529	2.4301	5.5274	2.2904	3.5608
2015-03-26	0.0	2.2849	3.529	2.4301	5.5274	2.2962	3.5608

TABLA 1. MSE

Obs.	AR		MA		ARMA	
	MAE.	R^2	MAE	R^2	MAE	R^2
0.0	1.8389	-896.4359	2.3404	-1404.25	1.8482	-904.2841

TABLA 2. R^2 , MAE

Si bien el modelo AR mostró una mejor predicción según las métricas, es crucial considerar el contexto en el que se desarrollaron estas estimaciones. Es importante destacar que el modelo ARMA solo utilizó 5 rezagos, mientras que el modelo AR empleó 39. A pesar de la diferencia en la cantidad de rezagos, el ajuste entre ambos modelos fue relativamente similar, lo que sugiere que el modelo AR podría superar al ARMA si se consideraran más rezagos.

3.9. Mejor Modelo Estadístico. A pesar de que las métricas indicaron un mejor rendimiento para el modelo AR, el modelo ARMA logró una estimación similar al modelo AR. Además, el ARMA alcanzó esta eficacia utilizando solo 5 rezagos, mientras que el AR necesitó 39 rezagos para lograr un rendimiento comparable. Esta diferencia en el ajuste es significativa y sugiere que, con un número similar de rezagos,

el ARMA podría superar al AR en términos de precisión. Por lo tanto, concluimos que el modelo ARMA es la mejor opción para la predicción de la precipitación en Honduras, ya que logra un buen ajuste con un número más bajo de rezagos.

3.10. Comparación Modelo Estadístico y el Modelo Redes Neuronales.

En un período de 61 días, las redes neuronales demostraron una notable precisión en la predicción de los datos. Sin embargo, conforme se extendió el horizonte temporal, su capacidad predictiva experimentó un declive considerable. Esto se evidenció especialmente al considerar su manejo de datos atípicos, donde su desempeño fue menos consistente. Aunque estas discrepancias son relevantes, es importante destacar que las redes neuronales lograron mantener una ventaja significativa sobre los modelos estadísticos en términos de precisión general.

Fecha	Valor Real	ARMA(1,3)	Redes Neur.
2015-01-24	0.0	3.5608	0.003819

TABLA 3. Comparación de AR(39) Vs RN | MSE

Al evaluar las métricas de desempeño, las redes neuronales mostraron una clara superioridad sobre los modelos estadísticos. Esto sugiere que, a pesar de los desafíos asociados con la volatilidad de los datos y la presencia de valores atípicos, las redes neuronales ofrecieron una capacidad predictiva más robusta y adaptable. Por lo tanto, se puede concluir que, en el contexto de la estimación de la precipitación en Honduras, las redes neuronales emergen como la opción preferida debido a su capacidad para capturar patrones complejos y no lineales en los datos, superando así las limitaciones de los modelos estadísticos tradicionales.

4. CONCLUSIONES

- Aunque el modelo AR demostró ser ligeramente superior en términos de métricas de comparación como MSE, R^2 , y MAE, la elección del modelo ARMA como la mejor opción entre los modelos estadísticos considerados se basa en su capacidad para lograr un rendimiento comparable al AR utilizando menos rezagos.
- Las Redes Neuronales Multicapa superaron en términos de precisión estimación al mejor Modelo estadístico. A pesar de los desafíos asociados con los datos atípicos y algunas variaciones, las redes neuronales demostraron una capacidad generalmente superior para prever los datos de precipitación, sugiriendo una habilidad para capturar patrones más complejos y no lineales en los datos.
- La precisión de la red neuronal disminuye gradualmente con el tiempo, lo cual puede atribuirse principalmente a la presencia de datos atípicos en la serie de tiempo. Estos datos atípicos pueden introducir ruido en el modelo y afectar su capacidad para generalizar correctamente a medida que avanza en el tiempo.
- Un factor importante que afectó el desempeño de todos los modelos fue la volatilidad inherente en los datos de precipitación. Esta volatilidad presentó dificultades para los modelos al tratar de capturar y predecir patrones en los datos, lo que contribuyó a los altos errores observados. Esta dificultad resalta la complejidad del problema de estimar la precipitación y sugiere la

necesidad de modelos más sofisticados y técnicas de modelado avanzadas para abordar eficazmente esta volatilidad.

- Para futuras investigaciones, se recomienda abordar la serie de tiempo con valores extremos centrándose en datos atípicos a partir de un umbral. Esto puede implementarse tanto en modelos estadísticos como en modelos de redes neuronales. Además, en cuanto a la estructura de redes neuronales, se necesita una estructura más compleja, como las redes neuronales recurrentes, que proporcionan un nivel de memoria para capturar mejor las relaciones temporales en los datos de precipitación.
- Se sugiere ampliar el enfoque hacia un modelo multivariado que incluya variables adicionales como el fenómeno del Niño, la temperatura y, de ser posible, la humedad del suelo. Estos factores son fundamentales en la alteración de las precipitaciones y su integración en el análisis proporcionaría una comprensión más completa de los patrones climáticos.

REFERENCIAS

1. Alexey Kaplan & Mark A. Cane & Yochanan Kushnir & Amy C. Clement & M. Benno Blumenthal & Balaji Rajagopalan. *Analyses of global sea surface temperature 1856–1991*. Journal of Geophysical Research: Oceans, 103(C9). 1998.
2. Arik, S. O., & Pfister, T. *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting*.
3. Cheung, Yin-Wong, and Kon S. Lai. *Practitioners corner: Lag Order and Critical Values of a Modified Dickey-Fuller Test*. Oxford Bulletin of Economics and Statistics 57.3 1995.
4. De Arce, Rafael, and Ramón Mahía. *Modelos Arima*. Programa CITUS: Técnicas de Variables Financieras 2003.
5. George E. P. Box, & Gwilyn M. Jenkins, & Gregory C. Reinsel *Time series analysis: forecasting and control*. Vol. 5 John Wiley & Sons. (2015)
6. Gulli, Antonio, and Sujit Pal. *Deep learning with Keras*. Packt Publishing Ltd, 2017.
7. Hsieh, W. W., Tang, B., & Strong, A. E. . *Use of self-organizing neural networks in the analysis and prediction of hurricane intensity*. (1995).
8. Kingma, Diederik P., and Jimmy Ba. "ADAM: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
9. Larranaga, Pedro, Inaki Inza, and Abdelmalik Moujahid. *Tema 8. redes neuronales*. Redes Neuronales, U. del P. Vasco 12 1997.
10. Liang, J., *A self-organizing map-based algorithm for detecting extreme climate events using spatiotemporal data*. International Journal of Climatology, 33(3), 698-710. 2013
11. Ramón, M. Dolores García. *Valor actual del modelo de Von Thünen y dos comprobaciones empíricas*. Revista de geografía 1976.
12. Peter J. Brockwell Richard A. Davis, *Introduction to Time Series and Forecasting* Vol. 2, Editorial Board 1996
13. Ovares, Rafael González. *El cambio climático en la agenda política: Un problema mundial*. Ingeniería 26.1 (2016): 59-70.
14. RUEY S. TSAY, *Analysis of Financial Time Series* Vol. 3
15. Song, Qiang, and Augustine O. Esogbue. *A new algorithm for automated Box-Jenkins ARMA time series modeling using residual autocorrelation/partial autocorrelation functions*. Industrial Engineering and Management Systems 5.2 2006.
16. Stock, James H. *Unit roots, structural breaks and trends*. Handbook of econometrics 4 1994.
17. Terceiro Lomba, Jaime. *Economía del cambio climático*. Anuales de la Real Academia de Ciencias Morales y Políticas. Ministerio de Justicia, 2008.
18. Vorobioff, Juan, et al. *Inteligencia Artificial y Redes Neuronales. Fundamentos, Ejercicios y Aplicaciones Con Python y Matlab*. 2022.
19. Wang, W., *Interpretable Deep Learning for Time Series Forecasting* Ecological Modelling, 2021

20. Wang, Yanjun, and Qun Liu. *Comparison of Akaike information criterion (AIC) and Bayesian information criterion (BIC) in selection of stock–recruitment relationships*. Fisheries Research 77.2 2006.
21. Yaglom, A. M. (1955). *The correlation theory of processes whose n -th difference constitute a stationary process*, 1995

ESCUELA DE MATEMÁTICA, UNIVERSIDAD AUTÓNOMA DE HONDURAS, TEGUCIGALPA, FRANCISCO MORAZÁN

AJUSTE Y CALIBRACIÓN DE CURVAS DE RENDIMIENTOS

ANDREW SEBASTIÁN CASTEJÓN RODRÍGUEZ

RESUMEN. La presente investigación se centra en el estudio comparativo de la estimación de la estructura temporal de la tasa de interés o más conocida como curva de rendimientos mediante la implementación de los modelos Nelson-Siegel (NS) y Nelson-Siegel-Svensson (NSS), utilizando no series temporales sino un enfoque basado en la selección de 24 fechas distintas durante el año 2023. Centrándose en la curva de rendimiento, se examinarán las capacidades predictivas y la adaptabilidad de estos modelos a la dinámica cambiante del mercado, adicionalmente, también se investiga como se diferencian en cuanto a su rendimiento.

ABSTRACT. The present research focuses on the comparative study of the estimation of the temporal structure of the interest rate or better known as the yield curve through the implementation of the Nelson-Siegel (NS) and Nelson-Siegel-Svensson (NSS) models, using not time series but an approach based on the selection of 24 different dates during the year 2023. Focusing on the yield curve, the predictive capabilities and adaptability of these models to the changing dynamics of the market will be examined, additionally, research will also be carried out how they differ in terms of performance.

1. INTRODUCCIÓN

En la gestión financiera y la actuaría existe un concepto llamado curvas de rendimientos, las cuales son usadas para realizar estrategias financieras y para marcar pautas en los mercados de diversos bienes y servicios. Es de gran interés para los bancos centrales, gobiernos y cualquier otro tipo de institución financiera, por lo que sus modelos de predicción y los conceptos que giran alrededor de ella son un tema de alto interés para cualquier profesional economista, financiero o actuario, pues con una mejor estimación estas estrategias se volverán mucho más sólidas. En este contexto, la presente investigación se adentra en la comparación de tres modelos prominentes para la estimación de curvas de rendimiento los modelos Nelson-Siegel y Nelson Siegel Svensson [7].

La volatilidad en los mercados financieros, las fluctuaciones en las tasas de interés y las variables macroeconómicas intrínsecas a cualquier país requieren una comprensión profunda y herramientas analíticas avanzadas para informar decisiones fundamentadas [4]. La curva de rendimiento, que representa la relación entre los rendimientos y los vencimientos de distintos instrumentos financieros, sirve como un indicador clave que refleja las expectativas del mercado y las condiciones económicas subyacentes [3].

Date: 05 de Febrero de 2024.

Key words and phrases. Modelo, Curva de Rendimiento, Estimación, Validación, Ajuste.

El modelo de Nelson-Siegel, un pilar en la estimación de curvas de rendimiento, ha demostrado su utilidad en diversos entornos, proporcionando una estructura flexible para capturar la pendiente y la curvatura de la curva, este modelo ha sido ampliamente utilizado por los analistas debido a su simplicidad y a que presenta consistencia entre la tasa forward y la curva de rendimiento [3]. Sin embargo, en la búsqueda de mayor precisión y adaptabilidad, otros modelos han emergido como competidores destacados [4]. El modelo de Svensson extiende la flexibilidad del modelo de Nelson-Siegel al incorporar términos adicionales.

En este contexto, el presente estudio se propone realizar una comparación de estos dos modelos, evaluando sus capacidades predictivas y su capacidad para adaptarse a las complejidades de los mercados financieros. La investigación se estructura en secciones que abordan aspectos clave de los modelos seleccionados, su fundamentación teórica, es decir, sus planteamientos matemáticos, la metodología de estimación de parámetros y, finalmente, un análisis comparativo de su rendimiento en los resultados.

2. ANTECEDENTES

Un bono es un certificado de endeudamiento público o privado que devenga intereses. Existe además otro concepto llamado bono de descuento el cual es un bono que no tiene cupones y tiene un único pago realizado en el momento del vencimiento [1]. Una letra del tesoro de los Estados Unidos es un ejemplo de este tipo de bonos, una vez definidos estos conceptos hay que analizar la historia sobre ellos, específicamente sobre lo que ocurrió en Estados Unidos para comprender la utilidad de las curvas de rendimiento.

El contexto histórico y teórico del control de la curva de rendimientos, comenzó con la contribución del economista John M. Keynes durante la Gran Depresión, quien propuso que los bancos centrales tenían la capacidad de influir en las tasas de interés a largo plazo para estimular la economía. Esta idea se puso en práctica notablemente durante la Segunda Guerra Mundial y en la Operación Twist en los años 60, donde se buscó aplanar la curva de rendimientos para reducir los costos del endeudamiento a largo plazo sin elevar las tasas a corto plazo.

La crisis financiera global de 2007-2008 marcó un antes y un después en la percepción y utilización del control de la curva de rendimientos. Los bancos centrales, enfrentando una economía en recesión y tasas de interés cercanas a cero, adoptaron medidas no convencionales, incluyendo la manipulación de la curva de rendimientos como herramienta para promover la estabilidad de precios y el crecimiento económico. Este enfoque resaltó la relevancia de la curva de rendimientos no solo como indicador económico, sino también como objetivo directo de la política monetaria, reafirmando la teoría de Keynes y adaptándola a los desafíos financieros contemporáneos [2].

El primer modelo de curva de rendimiento que se estará analizando en la investigación fue propuesto en 1987 por Charles B. Nelson y Andrew F. Siegel, quienes proponen un modelo de ajuste de la curva de rendimiento donde la tasa de retorno

depende de la madurez, es decir, el tiempo que tarda en vencer del instrumento, antes del modelo de Nelson y Siegel, las metodologías para modelar la curva de rendimientos incluían enfoques más simples, como el método de la media geométrica o la interpolación lineal, y modelos basados en la estructura a plazo de las tasas de interés que utilizaban supuestos simplificados sobre la forma de la curva [7]. También existían modelos que intentaban capturar la dinámica de la curva a través de regresiones polinómicas o mediante el ajuste de funciones exponenciales como se menciona unos párrafos más adelante; sin embargo no fue hasta la llegada de este modelo que se marcó un antes y un después.

El modelo surgió para estructurar una metodología que pudiera explicar de manera realista y sencilla el comportamiento de los rendimientos en el mercado de deuda, claro que estas consideraciones fueron dadas con argumentos ya que se había realizado un completo análisis de todos los modelos que, en su momento, eran reconocidos como los de mayor trascendencia técnica, y que de acuerdo con la percepción de Nelson y Siegel explicaban de una manera más apegada a la realidad la relación entre niveles de tasas de interés y plazos al vencimiento.

En el camino para generar un modelo, siguieron las notas de otros investigadores, uno de estos fue McCulloch quien en el año 1971 propuso aproximar la función del valor actual a través de un spline definido en tramos mediante polinomios vinculados a los datos de los precios del instrumento, Shea en el año 1984 demostró que la función de rendimiento tiende a doblarse bruscamente hacia el final del periodo de maduración, lo cual parece ser más una propiedad improbable que una verdadera relación con la curva de rendimiento. Nelson y Siegel llegaron a la conclusión de que era necesario proponer una metodología para la construcción de una curva de rendimiento fácilmente aplicable [4].

El modelo propuesto se basa en la solución de una ecuación diferencial ordinaria de segundo orden con raíces reales e iguales, la cual pertenece a las funciones conocidas como funciones de Laguerre [7].

Este modelo ha sido ampliamente utilizado por los analistas debido a su simplicidad y a que presenta consistencia entre la tasa forward y la curva de rendimiento. Diebold y Li en el año 2006 proponen una visión dinámica para este modelo estableciendo que Nelson-Siegel se compone de 3 factores, los cuales son: Nivel, Pendiente y Curvatura. El primero corresponde a la trayectoria de la tasa larga, mientras que el segundo está relacionado con el premio por plazo. Ambos factores son recuperables de los modelos de factores tradicionales. Sin embargo, la Curvatura parecía ajena a estos modelos. Christensen y otros en el 2009 presentan una derivación de este modelo bajo el paradigma de no-arbitraje, estableciendo que Nelson-Siegel puede recuperarse a través de los modelos de factores. En términos simples, la curvatura puede ser generada a través de un modelo de factores cuando uno de estos no es observable [4].

El modelo de Svensson, una extensión del modelo de Nelson y Siegel, fue introducido en 1994. Este modelo añade un término adicional a la fórmula original de Nelson y Siegel para aumentar la flexibilidad y permitir un mejor ajuste de la curva

de rendimientos, especialmente en entornos de mercado estresados. Esta adaptación permite identificar de manera más precisa las formas de la curva de rendimientos, incluyendo picos o valles más complejos [5].

3. ANÁLISIS Y MODELADO DE CURVAS DE RENDIMIENTO

3.1. Definiciones Previas. Las siguientes 3 definiciones provienen de [1]

Definición 3.1. Al tiempo actual, se define como el conjunto de las tasas de rendimiento de los bonos cupón cero de todos los vencimientos, $\{s_0(t)\}_{t>0}$, donde $s_0(t)$ es la tasa efectiva anual de rendimiento a partir del tiempo 0 para un bono cupón cero con vencimiento al tiempo t se le denomina estructura temporal de las tasas de interés; También es conocida como la curva de rendimiento de un bono cupón cero.

Más adelante se usará el acrónimo ETTI para referirse a esta definición.

Definición 3.2. El rendimiento al vencimiento de un bono cupón cero es llamado la tasa de interés al contado hasta el tiempo de vencimiento. En este sentido $s_0(t)$ sería la tasa de interés al contado (spot rate) para un bono cupón cero con vencimiento en o maduración en t años.

En este contexto el precio al tiempo 0 de un bono cupón cero es $\frac{1}{(1+s_0(t))^t}$, para un valor de vencimiento de 1.

Definición 3.3. Dada una estructura temporal para un bono cupón cero $s_0(t)$, en el momento 0, $n - 1$ año en adelante, la tasa de interés de un año para el año 0 desde el periodo $n-1$ hasta el periodo n , se denota como $i_0(n - 1, n)$, y cumple que:

$$1 + i_0(n - 1, n) = \frac{(1 + s_0(n))^n}{(1 + s_0(n - 1))^{n-1}}.$$

Otra relación que guardan ambas tasas es la siguiente:

$$(1 + s_0(n))^n = (1 + i_0(0, 1))(1 + i_0(1, 2)) \dots (1 + i_0(n - 1, n)).$$

Para más detalles y ejemplos del funcionamiento y la relación entre ambas tasas de interés vease la [1] cap 6, 315-321.

La notación general para la tasa de interés spot es $s_t(n)$, se refiere a la tasa spot de un instrumento de n años de maduración al tiempo t , de manera similar para la tasa forward sería $i_t(n - 1, n)$ y se refiere a una tasa efectiva anual de interés para el periodo de un año que transcurre entre $n - 1$ y n al tiempo t . El precio de un bono con cupones ($P_{n,t}$) depende de los flujos (F_i) que entregue, los cuales son descontados bajo sus respectivas tasas spot ($s_0(t_i)$) [3], Las tasas a plazo pueden calcularse utilizando la función de descuento o las tasas de interés al contado. Si las tasas de interés al contado son conocidas, entonces la ecuación del precio del bono se puede establecer como:

$$P_{n,t} = \sum_{i=1}^n \frac{F_i}{(1 + s_0(t_i))^{t_i}}$$

Donde F_i es el precio del i -ésimo cupón para un bono de n cupones, en el caso de un bono cupón cero, el precio se expresa como:

$$P_{0,t} = \frac{F}{(1 + s_0(t))^t}$$

Con F , siendo la cantidad fijada a pagar en el contrato del bono cupón cero[3].

Definición 3.4. El factor de descuento es el valor que refleja el cambio del valor presente de un instrumento respecto al tiempo.

en el contexto de un bono cupon 0 al tiempo t y de maduración n :

$$df_{n,t} = \frac{1}{(1 + s_t(n))^n}$$

$$\implies \frac{1}{df_{n,t}} = (1 + s_t(n))^n = (1 + i_t(0,1))(1 + i_t(1,2)) \dots (1 + i_t(n-1,n))$$

véase [6].

Esta definición es fundamental para comprender como surgen los modelos planteados más adelante.

Definición 3.5. Un arbitraje es una compra y venta simultánea de valores en diferentes mercados para beneficiarse de las discrepancias de precios [1].

Una transacción financiera que devuelve un monto positivo sobre una inversión de monto 0 sin riesgo es un ejemplo de arbitraje; es decir una inversión sin riesgo que tiene un rendimiento mayor que la tasa libre de riesgo.

Las siguientes definiciones fueron tomadas de [12].

Definición 3.6. El MSE o ECM por sus siglas en inglés es una medida comúnmente utilizada para evaluar la precisión de un modelo estadístico o de machine learning. Se calcula promediando los cuadrados de las diferencias entre los valores predichos por el modelo y los valores reales. Matemáticamente, se expresa como:

$$ECM = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2.$$

Definición 3.7. El MAE es otra medida de la precisión de un modelo, que calcula el promedio de las diferencias absolutas entre los valores predichos y los valores reales. Matemáticamente, se expresa como:

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

3.2. La Hipótesis de las Expectativas. Esta teoría propone que la estructura a término de las tasas de interés refleja las expectativas colectivas del mercado sobre las futuras tasas de interés. En otras palabras, bajo la hipótesis de expectativas, la tasa de interés a largo plazo es una proyección de las tasas de interés a corto plazo futuras.

En [3] se explica la construcción de la proposición que viene a continuación, la cual puede encontrarse en distintas literaturas como *"la hipótesis de las expectativas puras en logaritmo"*.

Proposición 3.8. *Dada la tasa de capitalización continua referente a la tasa spot $z_{n,t}$ y la tasa de rendimiento neto también capitalizada de manera continua $r_{n,t+1}$. La hipótesis de expectativas puras en logaritmo indica que, por arbitraje, el valor esperado de estos retornos, condicional a la información en t , debiera ser similar a los retornos de las estrategias de inversión segura. Estas corresponden a la compra de un bono de descuento que madure en el próximo período, lo cual significa que*

$$E_{t+i}(r_{n-i,t+1+i}) = z_{1,t+i}$$

lo que implica que:

$$z_{n,t} = \frac{1}{n} \sum_{i=0}^{n-1} E_t(z_{1,t+i}).$$

Para aclarar la notación vista en la anterior proposición con las propuestas en las definiciones previas $z_{n,t} = \ln(1 + s_t(n))$, por esto es que se le llama a $z_{n,t}$ como la tasa de capitalización continua, además $r_{n,t+1} = \ln\left(\frac{(1+s_t(n))^n}{(1+s_{t+1}(n-1))^{n-1}}\right) = \ln(i_{t+1}(n-1, n) + 1)$. La hipótesis de las expectativas puede ser usada en futuros trabajos para modelar las ETTI como series de tiempo.

3.3. Modelo de Nelson Siegel. El modelo original de Nelson-Siegel está definido en tiempo continuo donde la estructura de la tasa forward instantánea es modelada. Utilizando la proposición 3.7 en su versión continua los autores obtienen la curva de rendimiento, la cual depende de los parámetros de la tasa forward. Luego con datos efectivos de rendimiento de bonos de descuento estiman los parámetros. La modelación de la tasa forward permite una consistencia entre las tasas de descuento observadas y la directa interpretación de las expectativas de cambios en la tasa instantánea o de más corto plazo.

La motivación original del modelo es una estructura lineal con el parámetro no lineal calibrado. En aplicaciones empíricas típicamente se estima este parámetro en conjunto con los lineales.

Dado que el modelo es determinístico, el valor esperado de la proposición mencionada es innecesario [3]. Los autores proponen una función continua para describir la trayectoria de la tasa de interés forward instantánea en función de 4 parámetros [6].

$$f(n) = \beta_0 + \beta_1 \left(e^{-\frac{n}{\lambda}}\right) + \beta_2 \left(\frac{n}{\lambda} e^{-\frac{n}{\lambda}}\right).$$

Por la relación mencionada en las definiciones previas:

$$s_t(n) = \frac{\int_0^n f(s) ds}{n}$$

$$\implies s_t(n) = \beta_0 + \beta_1 \left(\frac{1 - e^{-\frac{n}{\lambda}}}{\frac{n}{\lambda}}\right) + \beta_2 \left(\frac{1 - e^{-\frac{n}{\lambda}}}{\frac{n}{\lambda}} - e^{-\frac{n}{\lambda}}\right).$$

Para un mayor desarrollo de las funciones y los planteamiento utilizados anteriormente véase [4]. Cada uno de sus parámetros tiene una interpretación económica específica que ayuda a explicar la forma de la curva de rendimientos. El parámetro β_0 se interpreta como la tasa de interés a largo plazo a la que convergen los rendimientos cuando el plazo tiende al infinito, β_1 captura la diferencia entre las tasas de interés a corto y a largo plazo, esencialmente reflejando la pendiente de la curva de

rendimientos en el extremo corto, β_2 representa el factor de curvatura de la curva de rendimientos, λ controla la tasa de decaimiento de los efectos de los factores β_1 y para el factor de β_2 un λ pequeño hace que la curva sea más sensible a cambios en los factores de pendiente y curvatura en plazos largos, mientras que un λ grande concentra estos efectos en el extremo corto de la curva, véase [10].

3.4. Modelo de Nelson Siegel Svensson. Svensson en el año 1994 propone una versión ampliada del modelo de Nelson Siegel (NS). La ecuación propuesta para la tasa forward instantánea en el período t , para un plazo de vencimiento de n , es la siguiente:

$$f(n) = \beta_0 + \beta_1 e^{-\left(\frac{n}{\lambda_1}\right)} + \beta_2 \left(\frac{n}{\lambda_1}\right) e^{-\left(\frac{n}{\lambda_1}\right)} + \beta_3 \left(\frac{n}{\lambda_2}\right) e^{-\left(\frac{n}{\lambda_2}\right)}$$

Entonces al hacer el mismo proceso anterior del modelo NS, se obtiene que la ecuación para la tasa spot viene dada por:

$$s_t(n) = \beta_0 + \beta_1 \left(\frac{1 - e^{-\frac{n}{\lambda_1}}}{\frac{n}{\lambda_1}}\right) + \beta_2 \left(\frac{1 - e^{-\frac{n}{\lambda_1}}}{\frac{n}{\lambda_1}} - e^{-\frac{n}{\lambda_1}}\right) + \beta_3 \left(\frac{1 - e^{-\frac{n}{\lambda_2}}}{\frac{n}{\lambda_2}} - e^{-\frac{n}{\lambda_2}}\right)$$

El modelo de Svensson es una extensión del modelo de Nelson-Siegel y añade un cuarto parámetro para capturar aún más la flexibilidad en la forma de la curva de rendimientos, especialmente en el tramo largo, β_3 es un segundo factor de curvatura introducido en el modelo de Svensson que permite una mayor flexibilidad en la parte larga de la curva de rendimientos, permitiendo modelar mejor las formas de las curvas de rendimiento que presentan comportamientos más complejos en sus extremos largos, λ_2 un segundo parámetro de decaimiento que controla específicamente el decaimiento del efecto del cuarto parámetro β_3 , permitiendo una mayor flexibilidad en la modelación de la parte larga de la curva de rendimientos, ver [11].

3.5. Supuestos de ambos modelos.

3.5.1. Forma Funcional Específica. El modelo presupone una forma funcional específica para la curva de rendimientos, compuesta por un término de nivel (largo plazo), un término de pendiente (corto plazo) y un término de curvatura (medio plazo). Esta forma funcional es suficientemente flexible para capturar una amplia gama de formas de curvas de rendimiento observadas en la práctica, pero el supuesto implica que todas las curvas de rendimiento pueden ser adecuadamente descritas por esta estructura.

3.5.2. Relaciones Lineales entre los Componentes. El modelo asume que la relación entre los rendimientos y los plazos puede ser capturada mediante la combinación lineal de tres componentes que dependen exponencialmente del plazo. Esto significa que, aunque el modelo puede capturar una variedad de formas de curvas de rendimiento, existen limitaciones en cuanto a la complejidad o características específicas de las curvas que puede modelar.

3.5.3. Independencia de los Errores. Como en muchos modelos de regresión, el modelo de Nelson-Siegel (y por extensión el modelo de Svensson) asume que los errores o residuos (las diferencias entre los rendimientos observados y los modelados) son independientes y distribuidos idénticamente. Esto implica que las desviaciones de la curva modelada no están correlacionadas a lo largo del tiempo o entre diferentes plazos.

3.5.4. Homocedasticidad. Se asume generalmente que los errores tienen varianzas constantes (homocedasticidad) a través de diferentes niveles de la variable independiente (en este caso, los plazos). Si los errores varían significativamente con el plazo, las estimaciones de los parámetros pueden ser ineficientes o sesgadas.

3.5.5. No Arbitraje. Aunque no es un supuesto directo del modelo en sí, en la práctica se asume que la curva de rendimientos refleja un mercado sin oportunidades de arbitraje, es decir, un mercado en el que no es posible generar ganancias sin riesgo explotando diferencias en los precios.

Es pertinente realizar una consideración sobre el supuesto de estacionariedad en el contexto de este estudio. Como es mencionado al principio del informe, el enfoque de la investigación no se orienta hacia el análisis de series temporales. El propósito primordial de esta investigación es el ajuste y validación de los modelos de NS y NSS a tasas de interés en instantes específicos, con el fin de examinar la configuración de la curva de rendimientos (es decir, su nivel, pendiente y curvatura) en estos puntos determinados. Por lo tanto, el análisis de estacionariedad no se considera un requisito previo para el ajuste del modelo. El tratamiento que se les está dando a estos modelos está específicamente diseñado para describir la estructura a plazo de las tasas de interés en momentos puntuales, sin que la estacionariedad de los datos constituya un factor relevante para su aplicación.

Los anteriores conceptos fueron tomados de [9].

3.6. Metodología de estimación. En los siguientes métodos, la estimación de los parámetros lineales, el vector de parámetros $\beta = (\beta_0, \beta_1, \beta_2)$ y en el caso del modelo de Nelson Siegel Svensson β_3 , se hace mediante mínimos cuadrados, lo que cambia es la manera en que se estiman los parámetros no lineales.

3.6.1. Mínimos cuadrados lineales con búsquedas incrementales (BI). el parámetro no lineal λ se estima mediante un proceso de búsqueda iterativa sobre un rango predeterminado de valores posibles para λ este proceso busca encontrar el valor de λ que, junto con los correspondientes valores de los parámetros lineales β_i , minimiza el error cuadrático total entre los rendimientos observados y los rendimientos modelados, en el caso del modelo de Svensson simplemente se realizan las iteraciones a la vez.

El primer paso para esta estimación consiste en definir un rango de búsqueda este rango debe ser lo suficientemente amplio como para incluir el valor verdadero de λ pero no tan amplio como para hacer ineficiente la búsqueda. Un ejemplo para esto podría ser, definir un máximo o un mínimo para ellos mediante aproximaciones por métodos numéricos. Luego se establece un incremento para la búsqueda dentro del rango definido. Este incremento determina qué tan fina será la búsqueda y cuántos valores de λ serán evaluados. Un incremento más pequeño aumenta la precisión pero también el tiempo computacional. Luego se procede a realizar las iteraciones y una vez completada la búsqueda incremental, se selecciona el valor de λ que minimizó el error cuadrático total [8]. Para la aplicación del método BI, los autores de [8] mencionan que estas se aproximan heurísticamente y recomiendan un mínimo de iteraciones, en base a ello se seleccionaron como parámetros:

3.7. Resultados. Para calibrar los modelos discutidos anteriormente se tomaron los datos del sitio oficial del departamento del tesoro de los Estados Unidos,

Modelos	NS	NSS	
Parámetros	λ	λ_1	λ_2
λ_{min}	0.01	0.01	0.01
λ_{max}	3	3	3
$\Delta\lambda$	0.01	0.05	0.05

estos datos corresponden a 24 fechas distintas del año 2023 y se pueden encontrar seleccionando como opción de búsqueda “Daily Treasury Par Yield Curve Rates”.

3.7.1. *Calibraciones obtenidas.* Las gráficas subsiguientes exhiben las calibraciones resultantes de la aplicación de dos modelos mediante el método de búsquedas incrementales, comparadas con los valores observados de la ETTI. Estos resultados corresponden a dos momentos específicos: la Figura 1 muestra los datos del 25 de mayo de 2023, mientras que la Figura 2 presenta los del 19 de octubre de 2023.

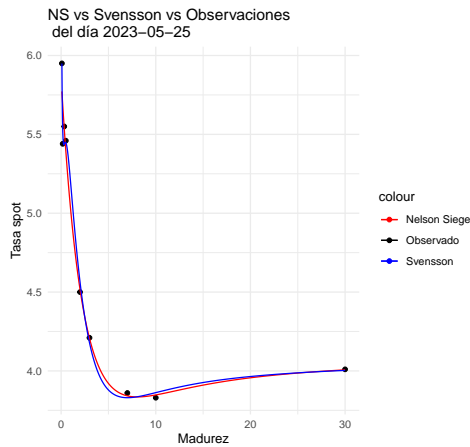


FIGURA 1. Para esa fecha el modelo de Nelson Siegel, obtuvo como calibraciones a $\beta_0 = 4.108$, $\beta_1 = 1.749$, $\beta_2 = -3.07$ y $\lambda = 2.31$ y en el de Svensson a $\beta_0 = 4.081$, $\beta_1 = 3.165$, $\beta_2 = -4.235$, $\beta_3 = -4.612$, $\lambda_1 = 1.750$ y $\lambda_2 = 0.1$.

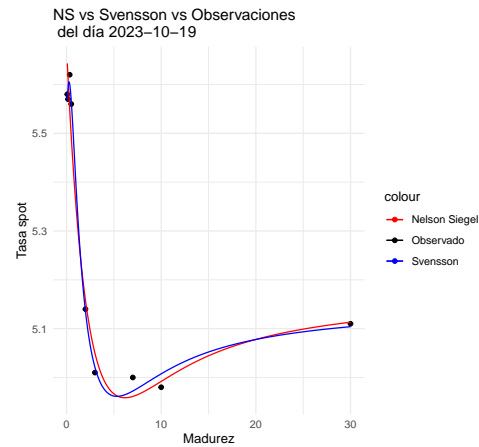


FIGURA 2. Para esa fecha el modelo de Nelson Siegel, obtuvo como calibraciones a $\beta_0 = 5.184$, $\beta_1 = 0.493$, $\beta_2 = -1.423$ y $\lambda = 2.29$ y en el de Svensson a $\beta_0 = 5.157$, $\beta_1 = 0.356$, $\beta_2 = 1.131$, $\beta_3 = -1.048$, $\lambda_1 = 0.35$ y $\lambda_2 = 2$.

A través de estas representaciones gráficas, se observa una tendencia consistente en la mayoría de las curvas ajustadas: los bonos con periodos de maduración más largos tienden a exhibir tasas de interés inferiores. Este fenómeno sugiere una inversión de la curva de rendimiento, lo que podría implicar expectativas de mercado de una disminución en las tasas de interés a largo plazo o reflejar una prima por el riesgo a mayor plazo.

Además, al comparar las curvas generadas por ambos modelos, se destaca la capacidad del modelo NSS para capturar dinámicas más complejas en la estructura a término, lo cual se evidencia en su ajuste más flexible a las variaciones de las

tasas a diferentes plazos, particularmente en los extremos de la curva donde el comportamiento de la tasa de interés se aparta del patrón predominante. Es importante señalar que si bien el ajuste de las curvas es notablemente preciso en las fechas seleccionadas, se requiere de un análisis más extenso para determinar la robustez de los modelos a lo largo de diferentes condiciones de mercado. En futuros trabajos, se podría ampliar este estudio para incluir series temporales completas que permitan una evaluación más exhaustiva del comportamiento dinámico de las tasas de interés y la efectividad de los modelos en la captura de choques económicos y financieros.

N°	Modelos Fechas	Nelson Siegel		Nelson Siegel Svensson	
		ECM	MAE	ECM	MAE
1	2023-01-18	0.007312	0.073235	0.000374	0.016128
2	2023-02-01	0.006447	0.065664	0.00029	0.013794
3	2023-02-15	0.001994	0.034792	0.000423	0.018218
4	2023-03-02	0.001158	0.027758	0.000541	0.015822
5	2023-03-16	0.003723	0.048261	0.002311	0.038019
6	2023-03-30	0.006105	0.061899	0.000807	0.02441
7	2023-04-13	0.008304	0.076476	0.003799	0.051945
8	2023-04-27	0.008198	0.076802	0.005022	0.062087
9	2023-05-11	0.049361	0.136698	0.01714	0.093425
10	2023-05-25	0.011094	0.063122	0.00325	0.04292
11	2023-06-09	0.005271	0.055541	0.000375	0.017497
12	2023-06-26	0.004057	0.040113	0.000255	0.009638
13	2023-07-11	0.002194	0.036044	0.0005	0.018778
14	2023-07-25	0.002437	0.035862	0.000271	0.013782
15	2023-08-08	0.005144	0.053912	0.000451	0.018999
16	2023-08-22	0.002471	0.041087	0.000375	0.016647
17	2023-09-06	0.002449	0.037947	0.000298	0.014651
18	2023-09-20	0.002604	0.042385	0.000661	0.02258
19	2023-10-04	0.003132	0.044252	0.000187	0.011903
20	2023-10-19	0.002085	0.03932	0.000326	0.015605
21	2023-11-02	0.002038	0.032964	0.000165	0.010911
22	2023-11-16	0.00084	0.024573	0.000371	0.017144
23	2023-12-01	0.001333	0.03007	0.000199	0.011579
24	2023-12-15	0.002488	0.03915	0.000219	0.012638

TABLA 1. La tabla proporcionada muestra los resultados del Error Cuadrático Medio (ECM) y el Error Absoluto Medio (MAE) para las calibraciones de los modelos Nelson-Siegel (NS) y Nelson-Siegel-Svensson (NSS) en 24 fechas diferentes durante el año 2023.

3.7.2. *Análisis Comparativo del Error Cuadrático Medio y el Error Absoluto Medio en la Estimación de la ETTI.* Una inspección detallada de la Tabla 1 revela patrones interesantes. Por ejemplo, para todas las fechas seleccionadas, el modelo NSS muestra consistentemente un ECM más bajo que el modelo NS, lo que sugiere una mejor capacidad de ajuste global. Este comportamiento es congruente con la naturaleza más flexible del modelo NSS, que, al incorporar parámetros adicionales,

es capaz de ajustarse más finamente a la forma de la curva de rendimiento. lo que es crítico en la aplicación práctica de la modelación de ETTI para la valoración de instrumentos financieros y la gestión de riesgos.

Los valores máximos de ECM y MAE para el modelo NS se presentan en las fechas 2023-05-11 y 2023-01-18 respectivamente, con el ECM más alto de 0.049361 y el MAE más alto de 0.136698. Por otro lado, el modelo NSS alcanza su ECM y MAE máximos en las fechas 2023-07-25 y 2023-05-11 respectivamente, con valores de 0.005271 para ECM y 0.093425 para MAE. Contrastando estos resultados con las gráficas previamente examinadas, es evidente que los modelos tienden a presentar mayores dificultades de ajuste en los extremos de la curva, especialmente en bonos con mayor madurez donde se observan tasas de interés más bajas. Este patrón se refleja en los valores más altos de ECM y MAE, indicando desviaciones más significativas entre las tasas de interés modeladas y las observadas para esos plazos.

3.7.3. Validación de supuestos. Para examinar la validez de los supuestos previamente delineados, se condujeron tres análisis estadísticos esenciales. El primero, la prueba de Ljung-Box, se empleó para evaluar la independencia de los residuos, anticipando un valor p que supera el umbral de significancia $\alpha = 1\%$, con un Lag ajustado basado en el $\ln(N^\circ$ de observaciones por modelo). Seguidamente, la prueba de Breusch-Pagan investigó la homocedasticidad de los residuos. Finalmente, la prueba de Shapiro-Wilks se utilizó para determinar la normalidad de los errores. Los resultados de estos análisis para los modelos de Nelson-Siegel y Nelson-Siegel-Svensson se ilustran en las figuras 3 y 4, respectivamente. para mayor información sobre las pruebas utilizadas véase [12].

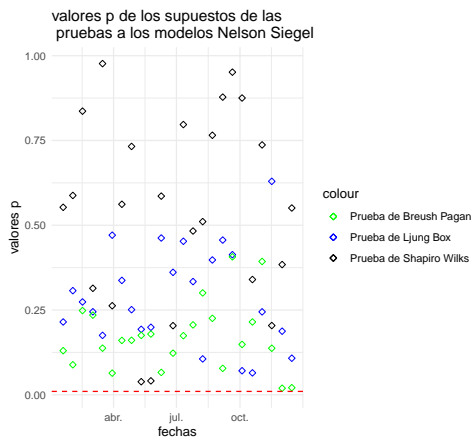


FIGURA 3. Representación gráfica de los modelos de Nelson Siegel que no rechazan H_0 de las 3 pruebas de los supuestos.

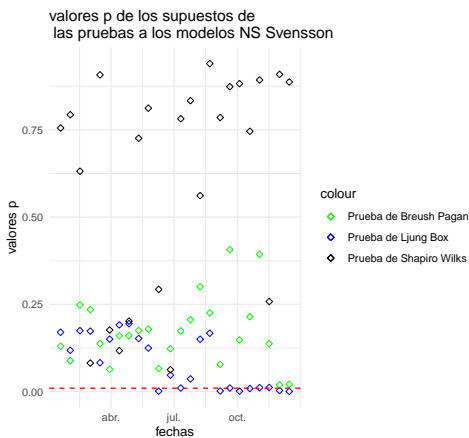


FIGURA 4. Representación gráfica de los modelos de NS Svensson que no rechazan H_0 de las 3 pruebas de los supuestos.

En el caso de los modelos Nelson-Siegel, los valores p derivados de las pruebas de Ljung-Box, Breusch-Pagan y Shapiro-Wilks reflejan una consistencia notable a

lo largo de las fechas seleccionadas para el análisis, permaneciendo por encima del límite de significancia de $\alpha = 1\%$. Este resultado sugiere que los residuos de los modelos cumplen con los criterios de independencia, homogeneidad de la varianza y normalidad, confirmando así la validez de los supuestos teóricos subyacentes. En contraste, los modelos Nelson-Siegel-Svensson muestran una variabilidad más marcada en los valores p a través de las distintas fechas evaluadas, aunque aún indican una aceptación general de los supuestos de independencia, homogeneidad de la varianza y normalidad de los residuos. Este patrón sugiere una adaptabilidad del modelo Nelson-Siegel-Svensson a las diferentes configuraciones de los datos de tasas de interés analizados.

3.7.4. *Validación cruzada.* A continuación las siguientes gráficas muestran el ajuste obtenido haciendo una validación cruzada, tomando como datos observados, a las tasas de interés correspondientes a las maduraciones de 3 meses, 1 año, 5 años y 20 años de las mismas fechas tomadas para el entrenamiento.

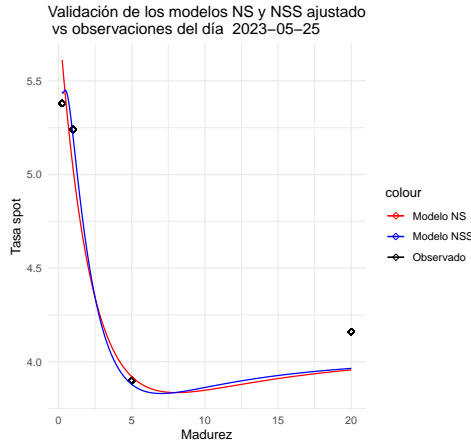


FIGURA 5. Validación y comparación de los modelos NS vs NSS del día 25 de mayo del 2023.

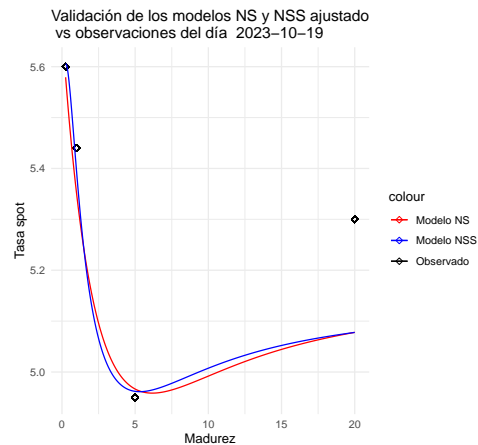


FIGURA 6. Validación y comparación de los modelos NS vs NSS del día 19 de octubre del 2023.

Se observa que las curvas modeladas por ambos NS y NSS siguen de cerca las tasas observadas en los horizontes de corto plazo (3 meses y 1 año), lo que indica una adecuada captura de la dinámica a corto plazo por parte de los modelos. Sin embargo, para horizontes de mayor madurez, especialmente en 20 años, las diferencias entre las tasas modeladas y observadas son más notables, sugiriendo un reto inherente al modelar comportamientos a largo plazo.

Los resultados de la validación cruzada sugieren que mientras el modelo NS proporciona un ajuste satisfactorio, el modelo NSS muestra una superioridad en la adaptación a los datos, reflejada en un ajuste más preciso a las tasas observadas en todas las madureces. Esta distinción es particularmente relevante en la práctica,

donde una predicción precisa a largo plazo es esencial para la valoración de instrumentos financieros y la gestión de riesgos.

A continuación se muestran gráficas representativas de los ECM y MAE obtenidos en la validación para ambos modelos, en las figuras 7 y 8 respectivamente.

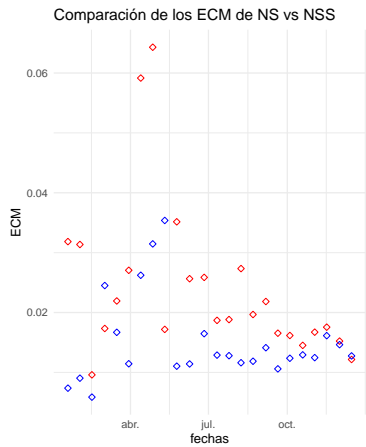


FIGURA 7. Comparación del ECM de los residuos de la validación cruzada para NS Y NSS.

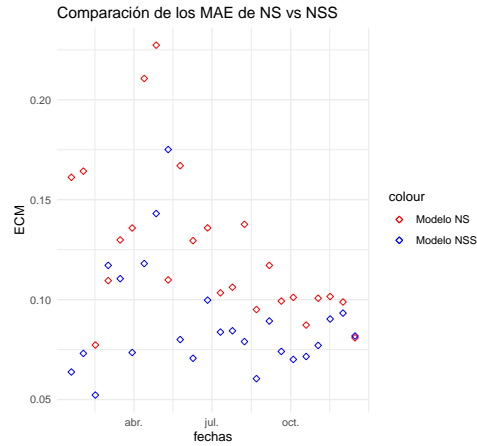


FIGURA 8. Comparación del MAE de los residuos de la validación cruzada para NS Y NSS.

Se puede observar que en general, el ECM y el MAE obtenido en los modelos NSS son menores a los de NS, lo cual refuerza el uso del modelo NSS y la estimación de los parámetros extras reflejados en la ecuación mostrada anteriormente, a pesar de tener un mayor costo computacional, pues el costo frente al modelo NS se eleva a causa del segundo parámetro no lineal que requiere de una anidación de ciclo de búsqueda extra.

4. CONCLUSIONES

4.1. Capacidad Predictiva y Adaptabilidad. Los resultados demuestran que el modelo NSS ofrece una mayor precisión y flexibilidad en la adaptación a las dinámicas del mercado en comparación con el modelo NS. Esta superioridad se evidencia en los menores valores de Error Cuadrático Medio (ECM) y Error Absoluto Medio (MAE) obtenidos para el modelo NSS en la mayoría de las fechas analizadas.

4.2. Efectividad en Diferentes Condiciones de Mercado. A través del análisis comparativo y la validación cruzada realizada, se ha observado que el modelo NSS es capaz de capturar con mayor fidelidad las complejidades y variaciones de la curva de rendimientos en distintos momentos, reflejando su utilidad en entornos de mercado variados.

4.3. Validación de Supuestos. La evaluación de los supuestos subyacentes a través de pruebas estadísticas para ambos modelos indicó que, en general, se cumplen las condiciones de independencia, homocedasticidad y normalidad de los residuos. Esto confirma la robustez y la validez teórica de los modelos NS y NSS para la modelación de curvas de rendimientos.

4.4. Direcciones Futuras. Se recomienda la realización de estudios adicionales que expandan el análisis a series temporales completas y que incluyan la evaluación del comportamiento de los modelos frente a choques económicos y financieros. Asimismo, la exploración de nuevas metodologías de estimación de parámetros podría ofrecer avances significativos en la precisión de la ETTI.

REFERENCIAS

1. A. Broverman, *Mathematics of Investment and Credit* 5. ed. ACTEX Publications Inc Winsted CT, 2010.
2. Amamiya M. , *History and Theories of Yield Curve Control*, Disponible en: https://www.boj.or.jp/en/about/press/koen_2017/ko170111a.htm, 2017. (Revisado en: 04 Marzo 2024).
3. Alfaro R., *La curva de rendimiento bajo Nelson-Siegel*. Documentos de Trabajo, (531), Banco Central de Chile, 2009.
4. J. Bedoya, L. Montes y H. Fernández, *Modelo de Charles B. Nelson y Andrew F. Siegel para la estimación de la estructura temporal de tasas de interés*. **39** (2018), no. 37, 1-10, ISSN 0798 1015.
5. Rigopoulos I., *Parametric Yield Curve Fitting to Bond Prices: The Nelson-Siegel-Svensson method*. Deriscope Blog. Disponible en: <https://blog.deriscope.com/index.php/en/yield-curve-excel-quantlib-nelson-siegel-svensson>, 2024. (Revisado en: 04 Marzo 2024).
6. J. Pereda, *Estimación de la Curva de Rendimiento Cupón Cero para el Perú*. Estudios Económicos, (17), 113-145. Banco Central de Reserva del Perú, 2009.
7. M. Choudhry, *Estimating and Fitting the Term Structure*. DOI: 10.1016/B978-075066263-5.50009-4., 2004.
8. Velásquez Giraldo, M., Gutiérrez Betancur, J. C., & Almonacid Hurtado, P. M. *Calibración de parámetros de los modelos de tasas de interés NS y NSS para Colombia: una nota técnica*. Journal of Economics, Finance and Administrative Science, **21**, 2016, 73–80. <http://dx.doi.org/10.1016/j.jefas.2016.06.003>
9. Wooldridge, J. M. *Introductory Econometrics: A Modern Approach*. 1. ed. Cengage Learning. South Melbourne, 2017.
10. Nelson, C. R., & Siegel, A. F. *Parsimonious Modeling of Yield Curves*. The Journal of Business, **60**(4), 473–489, 1987.
11. Svensson, L. E. O. *Estimating and Interpreting Forward Interest Rates: Sweden 1992-1994*. IMF Working Paper, WP/94/114, 1994.
12. G. James., D. Witten , T. Hastie , R. Tibshirani *An Introduction to Statistical Learning: with Applications in R*, 1. ed. Springer, 2013. DOI10.1007/978-1-4614-7138-7

DEPARTAMENTO DE MATEMÁTICA, UNIVERSIDAD AUTONOMA DE HONDURAS, TEGUCIGALPA, FRANCISCO MORAZÁN

Email address: andrew.castejon@unah.hn

MODELO DE REGRESIÓN MÚLTIPLE APLICADO A FACTORES QUE CAUSAN LA POBREZA EN HONDURAS

FREDYS ADALID CRUZ ROQUE

ABSTRACT. This article aims to show the professional in any area, the benefits of inferential statistics through the use of a special technique called Multiple Regression. To make use of this technique, it is necessary to evaluate and observe some important situations that cause the level of poverty, including lack of opportunities in their employment and the education in Honduras.

Resumen: Este artículo pretende mostrar al profesional de cualquier área, las bondades de la estadística inferencial mediante el uso de una técnica especial llamada Regresión lineal Múltiple. Para hacer uso de esta técnica, es necesaria de la evaluación y observación de algunas situaciones claves que causan el nivel de pobreza, entre ellas las faltas de oportunidades en el empleo y la educación en Honduras.

Date: Lunes 19 de febrero del 2024.

Key words and phrases. Regresión múltiple, Inferencia estadística, Medidas de tendencia .

1. INTRODUCCIÓN

En la mayoría de investigaciones, sin importar el campo de estudio en la cual se desarrollara, es necesaria la observación o experimentación para obtener datos de diferentes variables, por tanto, es fundamental determinar algún tipo de relación de dependencia entre variables con el objetivo de hacer predicciones a eventos, ya sean en el presente o futuro de acuerdo al comportamiento de las variables que se toman como objeto de estudio [13].

Hay diferentes situaciones donde se usa la regresión lineal múltiple por ejemplo el COVID-19, para implementar una vacuna fue necesario el estudio de dosis o cantidades de medicamento, para su aplicación en las personas, por lo cual fue necesario el uso de modelos matemáticos con variables referentes a este tratamiento. Por consiguiente, muchos profesionales confían en la intuición y la información que nos brinda el estudio de los métodos estadísticos para dar a conocer los resultados de las variables que se desean estudiar [3].

En esta investigación, se presenta el método de regresión lineal múltiple aplicado al análisis de la pobreza en Honduras. Para ello, se emplearán los indicadores clave que contribuyen significativamente a la deficiencia en la calidad de vida, como el desempleo, la falta de educación y exclusión social. A cada variable utilizada en la regresión lineal múltiple se le asignará una causa específica. Al aplicar este modelo, se esperan obtener resultados más precisos, permitiendo analizar cada dato, es decir, cada causa, con su respectivo porcentaje en el contexto del modelo de regresión lineal múltiple. Posteriormente, se realizarán observaciones detalladas para cada variable o causa.

Es importante destacar que la observación y los análisis de datos se llevarán a cabo con la ayuda de un lenguaje de programación llamado Python y con el programa Excel. Estas herramientas son fundamentales para ingresar los datos recolectados de cada variable y obtener información detallada sobre cada una de las variables relevantes en este tipo de problemas urbanos.

Existen numerosos trabajos que abordan temas relacionados con la planificación urbana, la cual busca destacar los problemas de una sociedad. Uno de estos trabajos, similar a esta investigación, se llevó a cabo en México, y se enfocó en investigar los factores demográficos, especialmente el considerable porcentaje de crecimiento en la migración [1].

Por lo tanto, un modelo de regresión lineal múltiple será fundamental para predecir el tipo de calidad de vida de las personas en cada uno de los departamentos, considerando tanto las causas favorables como las desfavorables en Honduras. Además, proporcionará información específica para realizar observaciones detalladas, permitiendo la mejor toma de decisiones favorables en la vida de las personas basadas en la investigación.

2. ANTECEDENTES

El término “regresión” se considera uno de los conceptos más importantes en estadística y fue utilizado por primera vez en el año 1877 por el estadístico Francis Galton (1822-1911). La aplicación inicial del método de regresión consistió en un estudio que reveló que la estatura de los niños nacidos de padres altos tiende a retroceder hacia la estatura media de la población. Por lo tanto, Francis Galton designó la palabra “regresión” al proceso general de predecir una variable, en este caso, la estatura del padre o de la madre [11].

Con el tiempo, los estadísticos acuñaron el término “regresión lineal múltiple” para describir el proceso en el cual se utilizan varias variables según el tema a investigar para predecir otras variables de interés.

Los principales desarrolladores del modelo de regresión lineal simple y de la regresión lineal múltiple que se destacaron por realizar sus aportes son muchos, dentro de los más mencionados según [10], son:

- (1) Francis Galton (1822–1911): Galton fue un científico británico que hizo contribuciones significativas al campo de la estadística. Introdujo el concepto de regresión hacia la media, fue un precursor importante de la regresión lineal.
- (2) Sir George Udny Yule (1871–1951): Yule, fue un estadístico británico, trabajó en el desarrollo de modelos de regresión lineal múltiple y es conocido por su trabajo en la regresión y la correlación.
- (3) Karl Pearson (1857–1936): Este estadístico británico fue pionero en el desarrollo de métodos estadísticos y contribuyó significativamente al establecimiento de la teoría de regresión.
- (4) Ronald A. Fisher (1890–1962): Fisher, fue un estadístico británico, hizo contribuciones fundamentales al diseño experimental y la inferencia estadística.

Desde el siglo XIX hasta hoy, el uso de la regresión lineal múltiple ha significado uno de los mayores aportes en matemáticas, aplicándose en diferentes áreas como economía, medicina, negocios e ingenierías. Al hablar de la regresión lineal múltiple, es necesario indicar el significado de sus variables para predecir factores como pobreza, tasas de interés, peso u otras variables según la investigación.

En la terminología de regresión se incluyen diferentes variables. La variable que se va a predecir se llama variable dependiente, variable a explicar o variable endógena, y la variable o las variables que se usan para predecir el valor de la variable dependiente se denominan independientes, explicativas o variables exógenas.

Entonces, para comprender el modelo de regresión lineal múltiple, es equivalente a decir que es una extensión del modelo lineal simple, utilizando más variables para analizar [14].

Además, es muy utilizado en diversos campos de investigación debido a su rapidez y facilidad de interpretación al realizar diferentes tipos de análisis en las variables. Hay diferentes áreas de investigación que han utilizado el modelo de regresión lineal múltiple para dar diferentes aportes a la sociedad. Dentro de las principales áreas y sus aportes según [9], están:

- (1) Análisis económico: Fundamental para la economía, modelando la relación entre múltiples variables como ingresos, tasas de interés, gastos, etc.
- (2) Medicina: Se aplica para evaluar la relación entre varios factores y resultados de salud, identificar factores de riesgo y prever resultados médicos.
- (3) Investigación en Ingeniería: Utilizado para analizar y prever el rendimiento de sistemas complejos, teniendo en cuenta múltiples factores.
- (4) Predicciones Financieras: Empleado para modelar y dar precios de acciones, rendimientos de inversiones y otros indicadores financieros, también en seguros y gestiones de riesgo.
- (5) Ciencias Ambientales: Utilizado para modelar la influencia de diversas variables en fenómenos como calidad del aire, temperatura, biodiversidad y energías renovables, proporcionando información esencial para la toma de decisiones y la gestión ambiental.
- (6) Análisis de Datos Biológicos: Aplicado en biología y genética para explorar cómo múltiples variables genéticas y ambientales contribuyen a fenotipos complejos.
- (7) Planificación Urbana: Puede ser utilizada para entender cómo diferentes variables impactan en el desarrollo urbano, incluyendo demografía, infraestructura urbana, sociología urbana, desarrollo sostenible urbano y tecnología para ciudades inteligentes .

Todas las áreas de investigación mencionadas anteriormente, donde se aplica la regresión, nos ayudan a reflexionar sobre las prioridades que tiene la estadística como rama de las matemáticas, con el modelo de regresión lineal múltiple se pueden identificar factores, como la educación, para evaluar la calidad de vida en Honduras, lo cual es esencial para determinar políticas que mejoren el bienestar de la población hondureña [8].

Ahora que se ha explicado la historia del concepto de regresión y se han mencionado sus principales exponentes, que con el paso del tiempo lograron identificar diferentes tipos de regresión, también se han dado a conocer las diferentes áreas de estudio donde el tipo de regresión lineal múltiple ha resuelto muchos problemas fundamentales. Por lo tanto, a continuación, se presentarán los conceptos fundamentales del modelo de regresión lineal múltiple y se aplicará a un problema de planificación urbana, donde se mostrarán los resultados obtenidos.

3. CONCEPTOS PRELIMINARES DE LA REGRESIÓN LINEAL MÚLTIPLE

3.1. Regresión lineal múltiple. La regresión lineal múltiple es uno de los temas fundamentales en estadística, considerada una técnica fundamental para predecir los valores de una variable dependiente basándose en el valor de dos o más variables independientes. Esto implica que, al utilizar este modelo, se asume que la relación entre las variables independientes y dependientes es lineal.

El modelo de regresión lineal múltiple está dado de la forma siguiente:

$$(3.1) \quad Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \dots + \beta_i X_{ii} + \epsilon_i$$

con $i=1,2,3,4,5,6, 7, \dots, n$

Donde los parámetros $\beta_0, \beta_1, \beta_2, \beta_3, \dots, \beta_i$ con $i=1,2,3,4,5,6, 7, \dots, n$ se puede decir que son los coeficientes desconocidos [13].

Ahora supongamos que tenemos una muestra aleatoria de tamaño “n”, por lo cual se puede dar a conocer el modelo de la regresión lineal múltiple de la siguiente forma:

$$(3.2) \quad Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \dots + \beta_k X_{ik} + \epsilon_i$$

con $i=1,2,3,4,5,6,7, \dots, n$ donde:

- Y_i : Es la variable dependiente (regresando o la variable a predecir)
- X_{ik} : Son las variables independientes (regresores, variable exógena o variable explicativa)
- β_k : Coeficientes de regresión (son parámetros fijos)
- ϵ_i : Errores o también se les denomina perturbaciones

Al expresar el modelo de regresión lineal múltiple para todas las observaciones de la muestra se obtiene un sistema de ecuaciones de la siguiente forma:

$$\begin{matrix} Y_1 = \beta_0 + \beta_1 X_{11} + \beta_2 X_{21} + \beta_3 X_{31} + \beta_4 X_{41} + \beta_5 X_{51} + \dots + \beta_k X_{n1} + \epsilon_1 \\ Y_2 = \beta_0 + \beta_1 X_{12} + \beta_2 X_{22} + \beta_3 X_{32} + \beta_4 X_{42} + \beta_5 X_{52} + \dots + \beta_k X_{n2} + \epsilon_2 \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ Y_n = \beta_0 + \beta_1 X_{1k} + \beta_2 X_{2k} + \beta_3 X_{3k} + \beta_4 X_{4k} + \beta_5 X_{5k} + \dots + \beta_k X_{nk} + \epsilon_n \end{matrix}$$

Luego las variables mencionadas anteriormente, se expresan en forma matricial de la siguiente forma:

$$(3.3) \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} X = \begin{bmatrix} 1 & X_{11} & X_{21} & X_{31} & X_{41} & X_{51} & \dots & X_{n1} \\ 1 & X_{12} & X_{22} & X_{32} & X_{42} & X_{52} & \dots & X_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & X_{1k} & X_{2k} & X_{3k} & X_{4k} & X_{5k} & \dots & X_{nk} \end{bmatrix} \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

A la matriz X se le llama matriz de regresores es decir es la matriz explicativa y dentro de la matriz X se incluye el regresor correspondiente al término independiente, conocido como regresor ficticio y su valor es de una unidad para cada observación que se va realizando.

Para realizar un análisis de regresión lineal múltiple se hacen las siguientes consideraciones sobre los datos y sus Condiciones para la regresión lineal múltiple, según [10]:

a) Linealidad: Se refiere a la relación lineal que se asume entre las variables independientes y la variable dependiente en el modelo de regresión. Esto significa que el efecto de cada variable independiente sobre la variable dependiente es constante y proporcional, es decir, un cambio en una unidad en una variable independiente produce un cambio constante en la variable dependiente, manteniendo constantes las demás variables independientes.

La forma más recomendable de comprobarlo es representando los residuos del modelo frente a cada uno de los predictores. Si la relación es lineal, los residuos se distribuyen de forma aleatoria entorno a cero. Estos análisis son solo aproximados, ya que no hay forma de saber si realmente la relación es lineal cuando el resto de predictores se mantienen constantes. Los valores de la variable dependiente están generados por el siguiente modelo lineal: $Y = X * \beta + \epsilon$.

b) Homocedasticidad: Se refiere a la suposición de que la varianza de los errores del modelo es constante para todos los valores de las variables independientes. En otras palabras, la homocedasticidad implica que la dispersión de los errores alrededor de la línea de regresión es constante en todos los niveles de las variables independientes, todos los errores tienen las mismas varianzas $V(\epsilon_k) = \sigma^2$.

c) Independencia: las perturbaciones aleatorias son independientes entre sí, es decir que $E(\epsilon_k \cdot \epsilon_l) = 0, \forall k \neq l$.

d) Normalidad: Se refiere a la suposición de que los errores del modelo siguen una distribución normal. En otras palabras, se asume que los residuos (diferencias entre los valores observados y los valores predichos por el modelo) tienen una distribución normal alrededor de cero, la distribución de la perturbación aleatoria tiene distribución normal $U \approx \mathcal{N}(0, \sigma^2)$.

e) Las variables explicativas X_k se obtienen sin errores de medida.

3.2. El método de mínimos cuadrados. La estimación de los parámetros por mínimos cuadrados en la regresión múltiple es el método utilizado para encontrar los coeficientes de regresión que mejor se ajustan a los datos observados. En la regresión múltiple, se busca encontrar una ecuación lineal que relacione una variable dependiente con dos o más variables independientes.

El método de mínimos cuadrados consiste en minimizar la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos por el modelo. En el caso de la regresión múltiple, se busca minimizar la suma de los cuadrados de los residuos, es decir, las diferencias entre los valores observados y los valores predichos por el modelo [3].

Matemáticamente, la estimación de los parámetros por mínimos cuadrados en la regresión múltiple se expresa como:

$$\beta = (X^T X)^{-1} X^T Y$$

Donde:

- β son los coeficientes estimados de la regresión.
- X es la matriz de diseño que contiene los valores de las variables independientes.
- Y es el vector de la variable dependiente.

Al realizar estos cálculos, se obtienen los coeficientes estimados que minimizan la suma de los cuadrados de los residuos, lo que permite encontrar la mejor línea de regresión que se ajusta a los datos observados en el modelo de regresión múltiple.

3.3. No colinealidad o multicolinealidad. Algo que es importante en la regresión múltiple es que los predictores deben ser independientes, ya que no debe haber colinealidad entre ellos. Se puede decir que la colinealidad ocurre cuando un predictor está linealmente relacionado con uno o varios de los otros predictores del modelo o cuando es la combinación lineal de otros predictores [5].

Una de las consecuencias de la colinealidad es que no se puede identificar de forma precisa el efecto individual que tiene cada una de las variables colineales sobre la variable respuesta, lo que indica que hay un incremento de la varianza de los coeficientes de regresión estimados hasta el punto que resulta prácticamente imposible establecer su significancia estadística.

Además, ocurren pequeños cambios en los datos que provocan grandes cambios en las estimaciones de los coeficientes. Si bien la colinealidad propiamente dicha existe solo si el coeficiente de correlación simple o múltiple entre algunas de las variables independientes es 1, esto raramente ocurre en la realidad. Sin embargo, es frecuente encontrar la casi-colinealidad o multicolinealidad no perfecta [6].

Para determinar la existencia de la colinealidad o multicolinealidad no existe un método estadístico concreto que encuentre multicolinealidad entre los predictores de un modelo de regresión, sin embargo, se han desarrollado numerosas reglas prácticas que tratan de determinar en qué medida afecta a la estimación y contraste de un modelo. Los pasos recomendados a seguir según [4]:

- Si el coeficiente de determinación es alto pero ninguno de los predictores resulta significativo, hay indicios de colinealidad.
- Se debe calcular una matriz de correlación en la que se estudia la relación lineal entre cada par de predictores. Algo que es importante tener en cuenta, que a pesar de no obtenerse ningún coeficiente de correlación alto, no está asegurado que no exista multicolinealidad. Se puede dar el caso de tener una relación lineal casi perfecta entre tres o más variables y que las correlaciones simples entre pares de estas mismas variables no sean mayores que 0.5.
- Al generar un modelo de regresión lineal simple entre cada uno de los predictores frente al resto. Si en alguno de los modelos el coeficiente de determinación R^2 es alto, estaría señalando a una posible colinealidad.
- La Tolerancia (TOL) y Factor de Inflación de la Varianza (VIF). Se trata de dos parámetros que vienen a cuantificar lo mismo (uno es el inverso del otro). El VIF de cada predictor se calcula según la siguiente fórmula:

$$(3.4) \quad VIF_{\beta_j} = \frac{1}{1 - R^2},$$

donde se obtiene de la regresión del predictor sobre los otros predictores. Esta es la opción más recomendada, los límites de referencia que se suelen emplear son:

- $VIF = 1$: Ausencia total de colinealidad
- $1 < VIF < 5$: La regresión puede verse afectada por cierta colinealidad.
- $5 < VIF < 10$: Causa de preocupación
- El término tolerancia indica parámetros específicos, por lo que los límites recomendables están entre 1 y 0.1.

En caso de encontrar colinealidad entre predictores, hay dos posibles soluciones. La primera es excluir uno de los predictores problemáticos intentando conservar el que, a juicio del investigador, está influyendo realmente en la variable respuesta. Esta medida no suele tener mucho impacto en el modelo en cuanto a su capacidad predictiva ya que, al existir colinealidad, la información que aporta uno de los predictores es redundante en presencia del otro. La segunda opción consiste en combinar las variables colineales en un único predictor, aunque con el riesgo de perder su interpretación [12].

Cuando se intenta establecer relaciones causa-efecto, la colinealidad puede llevar a conclusiones muy erróneas, haciendo creer que una variable es la causa cuando en realidad es otra la que está influenciando sobre ese predictor.

3.4. Análisis de varianzas. Se dice que la palabra ANOVA es el acrónimo de (Análisis de la Varianza)(Analysis of Variance en inglés). La tabla ANOVA, es una técnica estadística bastante utilizada para comparar las medias de tres o más grupos diferentes y determinar si hay diferencias significativas entre ellos [2].

ANOVA realiza diferentes tipos de trabajos y uno de ellos es descomponer la variabilidad total de los datos en componentes debidos a diferentes fuentes de variación, como el efecto de los grupos y el error aleatorio, permitiendo así determinar si las diferencias observadas entre los grupos son estadísticamente significativas.

Se utiliza bastante en el estudio de experimentos y estudios científicos para analizar la influencia de una o más variables independientes en una variable dependiente.

3.5. Normalidad. Cuando se habla de normalidad se habla de los análisis que se deben realizar en la normalidad, también llamados contrastes de normalidad, tienen como objetivo analizar cuánto difiere la distribución de los datos observados respecto a lo esperado, si procedieran de una distribución normal con la misma media y regresión típica. Existen diferentes pruebas para realizar el análisis de normalidad, entre ellas Shapiro-wilk y Jarque-Bera.

Shapiro-Wilk. La prueba de Shapiro-Wilk (SW) se restringió originalmente para tamaños de muestras pequeños ($n < 50$). Lo cual prueba que el primer intento que fue capaz de detectar desviaciones de la normalidad, ya es debido a la asimetría o curtosis, o ambas. Se ha convertido en la prueba preferida debido a sus buenas propiedades de potencia en comparación con un amplia gama de pruebas alternativas, entre ellas la prueba general de White, la cual no es necesario aplicar en este caso debido a que el tamaño de la muestra es pequeño para cada variable [7].

Las hipótesis a contrastar están establecidas de la siguiente manera:

H_0 : la muestra sigue una distribución normal .

H_1 : la muestra no sigue una distribución normal.

Shapiro y Wilk propusieron una prueba de bondad de ajuste dada en dos estimadores de desviación estándar:

L_n , el mejor estimador (MEL) bajo la hipótesis de que esté presente una distribución normal estándar, y el estimador de máxima verosimilitud (EMV) bajo la misma hipótesis.

El estadístico de prueba definido por Shapiro y Wilk en 1965 es el siguiente

$$(3.5) \quad W = \frac{(\sum_i^n a_i x_i)^2}{[\sum_{i=1}^n (X_i - \bar{x})^2]}$$

donde:

x_i : es la i -ésima estadística de orden.

\bar{x} : la medida de la muestra.

$$a_i = (a_1, \dots, a_n) = \frac{m^T V^{-1}}{(m^T V^{-1} m)^{1/2}}$$

$M = (m_1, \dots, m_n)^T$ son los valores esperados de las estadísticas de orden.

V : Es la matriz de covarianzas de las estadísticas de orden.

También se dice que la prueba de Shapiro-Wilk fue modificada por Royston en 1982 para ampliar la restricción del tamaño, cuyo propósito es habilitar en cálculo del estadístico W y su nivel de significancia para cualquier tamaño de muestra ($3 \leq n \leq 2000$).

Los valores estadísticos para la prueba de Shapiro-Wilk, recaen entre cero y uno. Para valores pequeños de la prueba de Shapiro-Wilk, hay evidencia de desviación de la normalidad, es decir se rechaza la normalidad; mientras que valor de uno indica la normalidad de los datos.

Ahora que se han presentado todos los conceptos fundamentales y necesarios para implementar el modelo de regresión lineal múltiple, se procederá a aplicar dicho modelo a uno de los principales problemas sociales que enfrenta Honduras.

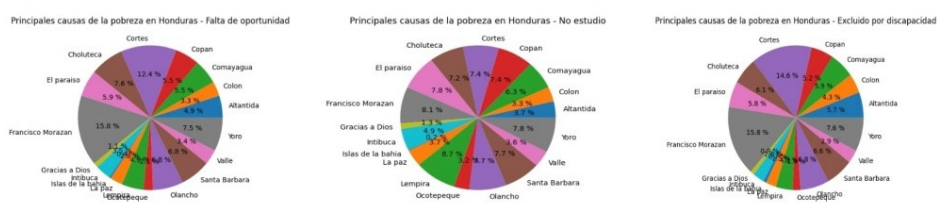
Se utilizará la base de datos con respecto a la pobreza en Honduras, la cual fue proporcionada por el Instituto Nacional de Estadísticas (INE) para realizar estudios de cada variable (factor). Entre estas variables se incluirá el factor de la falta de educación en el país y la falta de oportunidades de empleo.

4. MODELO Y RESULTADOS

La pobreza en Honduras es una de las causas sociales que más afecta al país, ya que es causada por diversos factores que van aumentando con el paso del tiempo. Según la Organización de Naciones Unidas (O.N.U.), Honduras está ubicada en el tercer puesto de los países más pobres a nivel centroamericano, y esto se debe al aumento de personas desempleadas.

A continuación, se presenta la base de datos elaborada mediante el uso de todas las encuestas que fueron realizadas por el Instituto Nacional de Estadísticas (I.N.E.) en cada uno de los departamentos en el año 2023. Esta base de datos muestra la cantidad de personas que no trabajan debido a los tres factores mencionados por la O.N.U. Para la elaboración de esta base de datos, se consideraron personas con edades comprendidas entre los 15 y 70 años.

Representación gráfica de los porcentajes para las Variables independientes:



Fuente : Elaboración propia.

Tabla 1. Base de Datos de la pobreza en Honduras

Departamento	Principales causas de la Pobreza en Honduras			Porcentaje
	No estudio	Falta de oportunidad	Excluido por discapacidad	
Altantida	41,213	161,694	18,614	4.68615
Colon	37,078	106,536	14,023	3.33472
Comayagua	71,499	181,022	19,175	5.74757
Copan	83,337	178,986	17,015	5.9092
Cortes	83,890	406,045	47,214	11.3630
Choluteca	81,059	247,322	19,720	7.36388
El paraiso	87,926	193,967	18,659	6.35800
Fco. Morazan	91,774	518,164	51,152	13.9849
Gracias a Dios	14,178	34,713	3,023	1.09821
Intibuca	55,061	114,116	8,924	3.76762
Islas de la bahia	1,927	12,020	2,233	0.34227
La paz	41,998	91,418	7,967	2.99087
Lempira	98,567	161,780	13,023	5.78298
Ocotepeque	36,367	68,415	6,111	2.34587
Olancho	86,579	222,584	22,138	7.00848
Santa Barbara	86,841	222,193	21,259	6.98716
Valle	40,320	109,975	9,531	3.38102
Yoro	88,175	244,100	24,520	7.54779
Total	1,127,789	3,275,050	324,301	100

Tabla 2. Modelo de regresión lineal múltiple:

	Coefficientes	Error típico	Estadístico t	Probabilidad	Inferior 95 %
intercepción	-4.3932e-10	9.47598e-1	-0.463615	0.650049	-2.47172E-09
Variable X ₁	2.11544e-05	2.24153E-14	94374983.6	4.96663E-119	2.115444E-05
Variable X ₁	2.11544E-05	2.08664E-14	1013802122	1.8225E-119	2.11544E-05
Variable X ₁	2.11544E-05	1.78559E-13	118472813.5	2.0576E-106	2.11544E-05

La ecuación lineal que mejor se ajusta al modelo de regresión lineal múltiple según los datos anteriores es: $Y = -4.3932 + 2.1154X_1 + 2.1154X_2 + 2.1154X_3$. Observa que el error típico cuantifica la variabilidad en cada prueba que se le dará a las variables X_1, X_2 y X_3 . El valor que se utiliza en pruebas de hipótesis y en intervalos de confianza para estimar los parámetros poblacionales se puede observar que es pequeño.

Tabla 3. Resultados para el coeficiente de determinación:

Estadísticos de Regresión	
Coefficiente	1
coeficiente	1
R^2 ajustado	0.99
Error típico	1.6398E-09
Observacion	18

En la tabla dos se indica que al realizar las 18 observaciones (departamentos), el R^2 explica la variabilidad de la variable dependiente que son los totales de porcentajes de las variables independientes incluidas en el modelo.

En otras palabras, un R^2 ajustado de 0.99 significa que una gran parte de las variaciones en la variable dependiente son explicadas por el modelo, pero no necesariamente que todas.

No implica que no haya necesidad de hacer ajustes en los datos para que se adapten a la base de datos. Un R^2 ajustado de 0.99 lo cual es indicativo de un buen ajuste del modelo a los datos, pero no necesariamente perfecto.

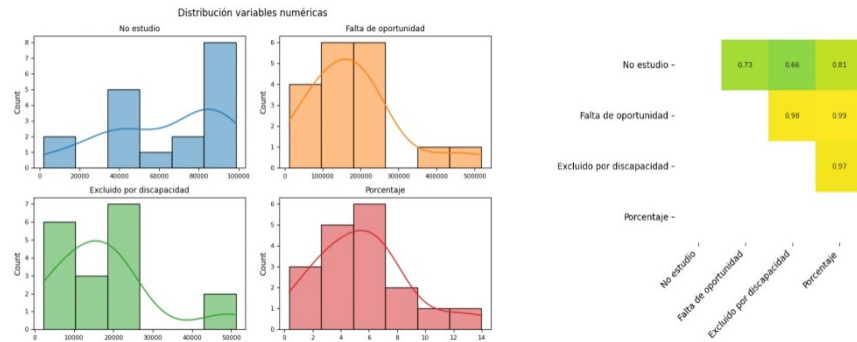
Tabla 4. ANOVA:

ANÁLISIS DE VARIANZA				
	Grados de libertad	Suma de cuadrados	Promedio de cuadrados	F
Regresión	3	194.57899417	64.85964722	2.41209E+19
Residuos	14	3.76452E-17	2.68894e-18	
Total	17	194.5789417		

valor crítico F
3.188e-131

Se sabe que la idea principal detrás del ANOVA es comparar la variabilidad entre grupos, con la variabilidad dentro de los grupos para determinar si existen diferencias significativas entre los grupos. Esto se logra mediante la comparación de las sumas de cuadrados entre y dentro de los grupos, utilizando los grados de libertad correspondientes para calcular la estadística F y realizar pruebas de significancia. Por lo tanto, se observa que el promedio de los cuadrados para la regresión es de 64.8596472, lo cual es diferente al valor de los residuos, que es de 2.6889 valor residual. Esto sugiere que las variables (factores) tienen diferencias significativas.

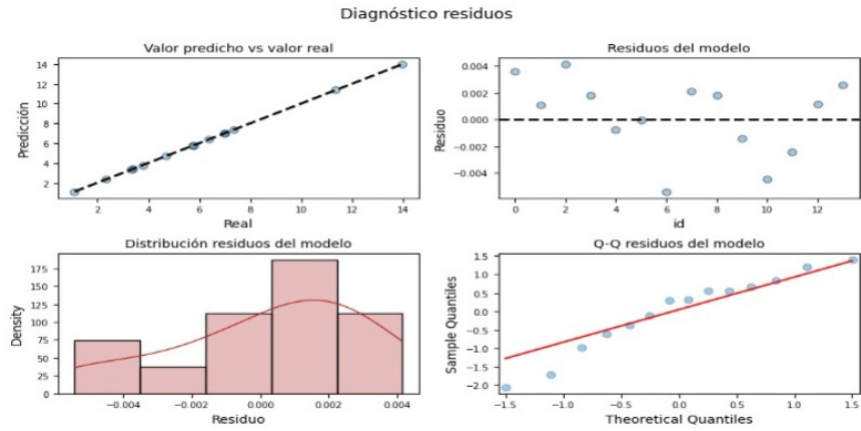
Figura 1. Grafico de la distribución normal :



Fuente : Elaboración propia

Del análisis preliminar se pueden extraer las siguientes conclusiones: Las variables que tienen una mayor relación lineal con el porcentaje de pobreza son la Falta de oportunidades con $r=0.98$ y la variable excluido por discapacidad con $r=0.97$, además la variable (no estudio) muestra una distribución exponencial. Por lo tanto, se concluye que al completar las variables para la normalidad, no se ve afectada la distribución de la variable que está indicando la cantidad de personas que no estudiarán.

Figura 2. Diagnóstico de Residuos:



Fuente : Elaboración propia

En la figura dos se observa que todos los residuos en cada gráfico se distribuyen de forma aleatoria en torno a cero, manteniendo aproximadamente la misma variabilidad a lo largo del eje X. Cuando se presenta este comportamiento de linealidad, indica que se cumple la normalidad y la homocedasticidad de los residuos.

Por lo tanto, se concluye que, mediante el análisis de la figura 1 y figura 2, se cumplió la normalidad, homocedasticidad y la colinealidad para el resultado de cada variable, por lo que no afectó la distribución que tiene cada una de las variables.

5. CONCLUSIONES

- (1) El modelo de regresión lineal múltiple es fundamental para el estudio de diferentes tipos de variables, lo cual hace más eficiente la resolución del problema que se está tratando de modelar y así obtener los resultados deseados.
- (2) Para generar una ecuación lineal que se ajuste al modelo de regresión lineal múltiple, se requiere la implementación de diferentes métodos matriciales, por lo cual, si se desea realizar los procedimientos en el menor tiempo posible, el uso de software es de gran ayuda.
- (3) Se observa que este modelo es bastante eficaz y también garantiza una mayor precisión en los resultados deseados. Por lo cual se ha podido determinar el comportamiento de cada variable, los resultados generales de la ecuación y resultarán correctos todos los supuestos de heterocedasticidad.

Como trabajo futuro, se debe considerar el uso de más factores (variables) que influyan en la pobreza de manera que todas las variables se integren en un modelo de regresión lineal múltiple más amplio. Esto permitirá encontrar una ecuación que incluya múltiples variables y pueda agrupar todos los diferentes tipos de pobreza, así como realizar predicciones más precisas. Además de obtener resultados, también se tienen que proponer estrategias socioeconómicas que reduzcan los indicadores de baja calidad de vida en cada habitante de la población hondureña.

REFERENCES

- [1] Marina Ariza and Luis Felipe Jiménez Chaves, *Selectos pero desiguales: inmigrantes latinoamericanos en México (1990-2015)*, Si Somos Americanos **21** (2021), no. 2, 170–202.
- [2] María Belén Castañeda, *Procesamiento de datos y análisis estadísticos utilizando spss: Un libro práctico para investigadores y administradores educativos*, Edipucrs, 2010.
- [3] Alfredo de Oliveira Pereira, *Medidas de tendência central*, Arquivos Brasileiros de Psicotécnica **5** (1953), no. 1, 61–70.
- [4] Alfonso GARCÍA PÉREZ, *Estadística aplicada avanzada con r*, Editorial UNED, 2022.
- [5] Alberto Ibarra Mares, *Desarrollo del análisis factorial multivariable aplicado al análisis financiero actual*, Eumed. net, 2010.
- [6] Lorena López Morán and José Hernández Alonso, *Estadística descriptiva*, Sanz y Torres, 2019.
- [7] Rosa María Nava Rogel and Patricia Mercado Salgado, *Evaluación de la calidad métrica para indicadores de capital intelectual generados a partir de bases estadísticas*, Revista de la educación superior **39** (2010), no. 155, 99–120.
- [8] Athanasios Papoulis, *Probability and statistics*, Prentice-Hall, Inc., 1990.
- [9] Ximena Aracely Rojas Aguirre, *Implementación de un sistema de recomendación de habitabilidad en la ciudad de Quito orientado a la planificación urbana cognitiva aplicando técnicas de machine learning* (2022).
- [10] Teddy Seidenfeld, *Philosophical problems of statistical inference: Learning from ra fisher*, Vol. 22, Springer Science & Business Media, 1979.
- [11] Laura Suárez, López Guazo, and Rosaura Ruiz Gutiérrez, *Eugenesia, herencia, selección y biometría en la obra de Francis Galton*, Llull: Revista de la Sociedad Española de Historia de las Ciencias y de las Técnicas **25** (2002), no. 52, 85–108.
- [12] Mario F Triola, *Estadística*, Pearson educación, 2004.
- [13] Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Ricardo Cruz, *Probabilidad y estadística*, Vol. 624, McGraw-Hill México, 1992.
- [14] Hanwen Zhang and Hugo Andrés Gutiérrez Rojas, *Teoría estadística: aplicaciones y métodos*, Hugo Andrés Gutiérrez Rojas, 2010.