

## Proyecto-Redes- Neuronales

### Keywords:

Artificial Neural Networks,  
Kolmogorov-Arnold Networks, Market  
Prediction

\*Author for correspondence. Email:  
zs23000636@estudiantes.uv.mx

# Comparación de métodos de IA en el análisis de series de tiempo del mercado financiero

Barrios B., Carlos A.\*

Instituto de Inteligencia Artificial, Universidad Veracruzana, Xalapa, Veracruz, México

### Abstract

En este trabajo se presenta una comparación exhaustiva entre tres modelos distintos de redes neuronales aplicados a la predicción del mercado financiero: la red neuronal artificial (ANN), la red neuronal convolucional (CNN) y la red Kolmogorov-Arnold (KAN). Utilizando datos históricos de los mercados de Google, Amazon y Apple obtenidos de Yahoo Finance, se entrenaron y probaron estos modelos para predecir las características de precios de apertura, cierre, mínimo y máximo.

El estudio demuestra que la KAN, con sus funciones de activación aprendibles y estructura basada en splines, proporciona resultados significativamente mejores en términos de precisión de predicción, en comparación con las tradicionales ANN y CNN. A través de una metodología rigurosa que incluye preprocesamiento de datos como la eliminación de tendencia y el escalado, se lograron mejoras notables en las métricas de error cuadrático medio (MSE) y error absoluto medio (MAE).

Los resultados indican que la KAN, especialmente utilizando splines cuadráticos y cúbicos, supera a las otras arquitecturas en la mayoría de los casos, sugiriendo que esta técnica tiene un gran potencial para aplicaciones futuras en el campo de la predicción financiera. Este trabajo destaca la importancia de la elección de modelos y la adecuación de sus parámetros para mejorar la precisión y robustez en las predicciones de mercado.

### Introducción

A finales del siglo XX, se iniciaron los primeros estudios para predecir los mercados de valores, esto tomando ventaja con los métodos de inteligencia artificial. Para comenzar a analizar los mercados, se necesita conocer por dos estrategias: *análisis técnico* y *análisis fundamental*. Mokhtari, Yen, and Liu 2021

Análisis técnico trata sobre cómo los accionistas intentan evaluar los mercados respecto a los datos históricos y sus indicadores. Un modelo de Machine Learning (ML) puede hacer lo anterior, se entrena para encontrar un patrón lógico entre los indicadores y el precio de cierre de las acciones; esto conduce a llevar un modelo de predicción que estime el precio de las acciones al final de un día, una semana, mes o año. Si el valor estimado del precio es mayor que el precio de las acciones, los accionistas reciben una señal de venta, mientras que si el precio estimado es menor que el precio de las acciones, reciben una señal de retención/compra. Es evidente que cualquier cambio en el informe financiero de una empresa puede afectar de inmediato el sentimiento público en las noticias y las redes sociales. Mokhtari, Yen, and Liu 2021

Otro modelo de ML puede investigar las noticias y las redes sociales a través de Internet para predecir un impacto positivo/negativo en los indicadores fundamentales de los precios de las acciones. Luego, proporciona la señal de acción para los accionistas basada en el sentimiento público.

En resumen, ¿Pueden los métodos de IA predecir/vencer al mercado financiero?

Para contestar lo anterior, se hará la comparación de 3 modelos distintos proporcionados por Mukherjee et al. 2021 donde intentan predecir algunas características usando dos modelos ANN (artificial neural network) y CNN (convolutional neural network). En este trabajo se propone el modelo de *Kolmogorov-Arnold Network* como otra alternativa para generar una predicción competitiva contra las clásicas y eficientes ANN. Para hacer la comprobación se usarán mercados populares del extranjero, Google (GOOG), Amazon (AMZN) y Apple (AAPL), usando las características del artículo principal referenciado Mukherjee et al. 2021 que son: 'Low', 'High', 'Open' y 'Close'.

## Marco teórico

### Propagación hacia atrás en la red neuronal de retroalimentación

Las redes neuronales representan el mapa no lineal  $F(X)$  sobre un espacio de entrada de alta dimensión utilizando capas jerárquicas de abstracciones. Un ejemplo de una red neuronal es una red de alimentación hacia adelante (feedforward), una secuencia de  $L$  capas formadas mediante composición Coqueret 2021:

Una red profunda de alimentación hacia adelante es una función de la forma:

$$\begin{aligned}\hat{Y}(X) &:= F_{W,b}(X) \\ &:= \left( f_{W_{(L)},b_{(L)}}^{(L)} \dots \circ f_{W_{(1)},b_{(1)}}^{(1)} \right) (X),\end{aligned}\quad (1)$$

donde:

- $f_{W_{(l)},b_{(l)}}^{(l)} := \sigma^{(l)}(W^{(l)}X + b^{(l)})$  es una función semi-afín, donde  $\sigma^{(l)}$  es una función de activación no lineal univariante y continua como  $\max(\cdot, 0)$  ó  $\tanh(\cdot)$ .
- $W = (W^{(1)}, \dots, W^{(L)})$  y  $b = (b^{(1)}, \dots, b^{(L)})$  son matrices de pesos y desplazamientos (también conocidos como sesgos), respectivamente.

Las funciones de activación son esenciales para que la red aproxime funciones no lineales.

Las raíces teóricas de las redes neuronales de alimentación hacia adelante están dadas por el teorema de representación de Kolmogorov-Arnold de funciones multivariadas. De manera notable, mostraron cómo las redes neuronales, con una capa oculta, son aproximadores universales de funciones no lineales Coqueret 2021.

### Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) son redes neuronales de alimentación hacia adelante que pueden explotar las estructuras espaciales locales en los datos de entrada. Las CNN intentan reducir el tamaño de la red aprovechando la localidad de los datos.

Las CNN profundas, con múltiples convoluciones consecutivas seguidas de funciones no lineales, han demostrado ser inmensamente exitosas en el procesamiento de imágenes. Se puede ver las convoluciones como filtros espaciales que están diseñados para seleccionar un patrón específico en los datos Coqueret 2021. Por esta razón, la convolución se utiliza con frecuencia para el procesamiento de imágenes. En el modelado financiero, generalmente se tiene diferentes estructuras espaciales, como las profundidades del libro de órdenes límite o la superficie de volatilidad implícita de los derivados. Sin embargo, la CNN también ha establecido su lugar en el análisis de series de tiempo, en Mukherjee et al. 2021 usan las CNN en histogramas 2-D creados a partir de las series de tiempo, de

tal manera se puedan usar y dar resultados aproximados la serie de tiempo real.

La convolución 2-D se trata de aplicar una pequeña matriz de núcleo  $K \in \mathbb{R}^{2k+1 \times 2k+1}$ , sobre la entrada de la matriz,  $X \in \mathbb{R}^{m \times n}$ , para dar una imagen filtrada.  $Y \in \mathbb{R}^{m-2k \times n-2k}$ . En el contexto de las CNN, los elementos de la imagen filtrada son referidos como los valores del mapa de características y son calculados de acuerdo con Coqueret 2021:

$$\begin{aligned}y_{i,j} &= [K * X]_{i,j} \\ &= \sum_{p,q=-k}^k K_{k+1+p,k+1+q} x_{i+p+1,j+q+1}, \\ i &\in \{1, \dots, m\}, j \in \{1, \dots, n\}.\end{aligned}\quad (2)$$

### Redes de Kolmogorov-Arnold

Mientras que las MLPs están inspiradas en el teorema de aproximación universal, las Redes Kolmogorov-Arnold (KANs) se inspiran en el teorema de representación Kolmogorov-Arnold. Al igual que las MLPs, las KANs tienen estructuras completamente conectadas. Sin embargo, mientras que las MLPs colocan funciones de activación fijas en los nodos, las KANs colocan funciones de activación aprendibles en las aristas ("pesos"). Como resultado, las KANs no tienen matrices de pesos lineales en absoluto: en su lugar, cada parámetro de peso se reemplaza por una función unidimensional aprendible parametrizada como una "spline" 6. Los nodos de las KANs simplemente suman señales entrantes sin aplicar ninguna no linealidad Liu et al. 2024.

Las KANs no son más que combinaciones de splines. Los splines son precisos para funciones de baja dimensión, fáciles de ajustar localmente y capaces de cambiar entre diferentes resoluciones. Sin embargo, los splines tienen un grave problema de maldición de la dimensionalidad (COD) debido a su incapacidad para explotar estructuras compositivas. Las MLPs, por otro lado, sufren menos de la COD gracias a su aprendizaje de características, pero son menos precisas que los splines en dimensiones bajas, debido a su incapacidad para optimizar funciones univariadas. Para aprender una función con precisión, un modelo no solo debe aprender la estructura compositiva. Las KANs son tales modelos, ya que tienen MLPs en el exterior y splines en el interior. Como resultado, las KANs no solo pueden aprender características, sino también optimizar estas características aprendidas con gran precisión Liu et al. 2024.

Vladimir Arnold y Andrey Kolmogorov establecieron que si  $f$  es una función continua multivariable en un dominio acotado, entonces  $f$  puede escribirse como una composición finita de funciones continuas de una sola variable y la operación binaria de adición.

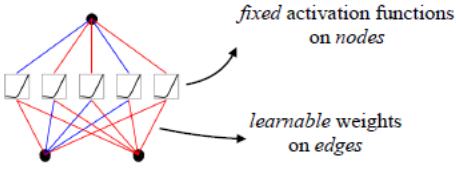
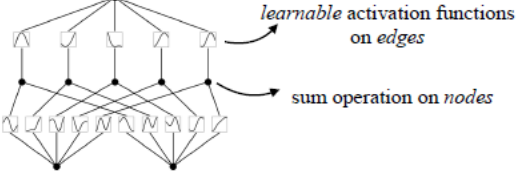
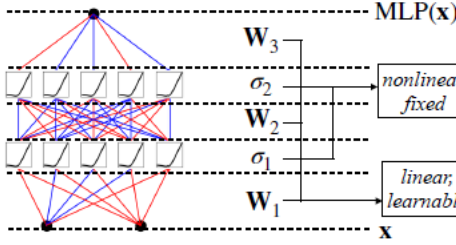
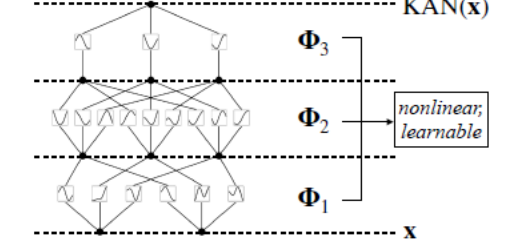
Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(e)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^n \phi_{q,p}(x_p) \right)$
Model (Shallow)	(a) 	(b) 
Formula (Deep)	$\text{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\text{KAN}(\mathbf{x}) = (\Phi_3 \circ \Phi_2 \circ \Phi_1)(\mathbf{x})$
Model (Deep)	(c) 	(d) 

Figure 1. Imagen obtenida de: Liu et al. 2024. Comparación de los modelos MLP contra KAN.

Más específicamente, para una función suave  $f : [0, 1]^n \rightarrow \mathbb{R}$ ,

$$f(\mathbf{x}) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^n \phi_{q,p}(x_p) \right), \quad (3)$$

donde  $\phi_{q,p} : [0, 1] \rightarrow \mathbb{R}$  y  $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$ . Liu et al. 2024

Las capas de KANs con entrada de dimensión  $n_{in}$  y salidas de dimensión  $n_{out}$  se pueden definir como una matriz de funciones unidimensionales

$$\Phi = \{\phi_{q,p}\}, \quad p = 1, 2, \dots, n_{in} \quad q = 1, 2, \dots, n_{out},$$

donde las funciones  $\phi_{q,p}$  tienen parámetros entrenables. En el teorema de Kolmogorov-Arnold, las funciones internas forman una capa de KAN con  $n_{in} = n$  y  $n_{out} = 2n + 1$ , y las funciones externas forman una capa de KAN con  $n_{in} = 2n + 1$  y  $n_{out} = 1$ . Así que las representaciones de Kolmogorov-Arnold en la ecuación 3 son simples composiciones de dos capas KAN. En pocas palabras, que sea más profunda una KAN es proporcional a decir que se apilan más capas KAN.

#### Base de datos

La base de datos a trabajar fueron tomados por **Yahoo Finance** Yahoo Finance, donde se obtuvieron los datos de las

características 'Open', 'Close', 'Low', 'High', de aquí se hizo en el intervalo de tiempo de '2014-01-01' a '2024-04-01' con pasos de un día. Los mercados que se usaron fue el de Amazon (AMZN), Google (GOOG) y Apple (AAPL).

#### Modelo propuesto

Como ya se había mencionado, este proyecto se hizo con base al artículo proporcionado por Mukherjee et al. 2021, por ello, se siguieron sus dos modelos tal cual lo hacen en su artículo, sin embargo, se hará unas cuantas modificaciones para optimizarlos y evitar el sobreajuste, además se adiciona un nuevo método como otra alternativa a seguir.

#### Modelo de ANN

La estructura de este red es tener 8 neuronas en su capa de entrada, como se hace la predicción de los días  $(n-1)^{th}$  y  $(n-2)^{th}$ , es decir del anterior y el anterior del anterior y son 4 características, nos da 8. Tendrá dos capas intermedias con 2 neuronas en cada una y por último, la capa de salida tendrá 4 neuronas correspondientes a las 4 características. Tendrá el algoritmo de retropropagación con puras funciones de activación sigmoide, sin embargo, también se probará con ReLu y Lineal para comprobar la efectividad de lo propuesto en el artículo. Su tasa de aprendizaje en el artículo es  $\eta = 0.1$ , pero en este proyecto se hará de  $\eta = 0.01$ . No se entiende el por qué tener una tasa muy alta, siempre se usan valores menores a 0.01. Y por último la función de costo será la métrica MSE (Mean Squared Error).

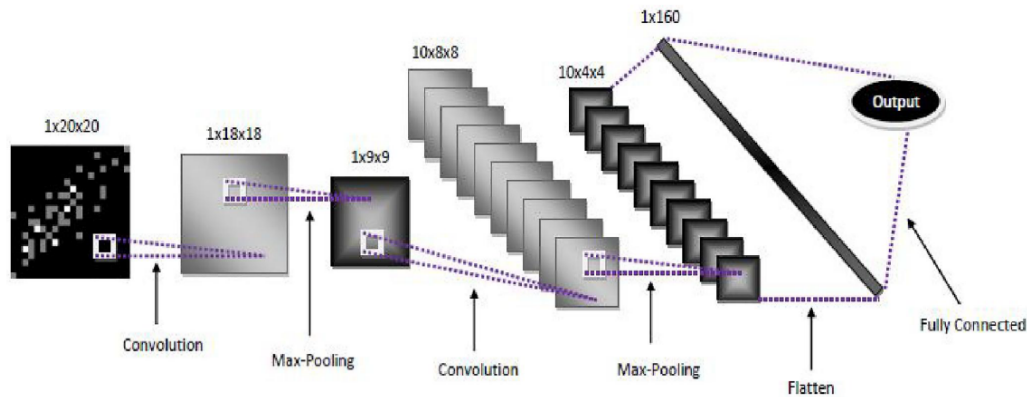


Figure 2. Imagen obtenida de: Mukherjee et al. 2021. Estructura de la red neuronal convolucional.

### Modelo de CNN

Para este modelo se necesitan de imágenes para que la red convolucional pueda leer los datos e intentar predecir con ella. La preparación de los datos se explica mejor en la siguiente sección.

Como se observa en la figura 6, se tiene la siguiente estructura de la CNN:

```
# 1 filtro, kernel 3x3
Conv2D(1, (3, 3), activation='relu',
input_shape=(20, 20, 1)),
# Max-pooling 2x2
MaxPooling2D(pool_size=(2, 2)),
# 10 filtros, kernel 2x2
Conv2D(10, (2, 2), activation='relu'),
# Max-pooling 2x2
MaxPooling2D(pool_size=(2, 2)),
# Aplana el tensor resultante
Flatten(),
# Capa fully connected con 160 neuronas
Dense(160, activation='relu'),
# Capa de salida
Dense(1, activation='linear')
]
```

Se usó el optimizador 'Adam' con 5000 épocas y una tasa de aprendizaje de 0.01. Por último, se ocupó la misma función de pérdida que ANN que es MSE.

### Modelo de KAN

Para este modelo se hizo uso de la paquetería hecha con base al artículo de KAN Liu et al. 2024, la cual está hecha en python. Los únicos hiperparámetros que se pueden modificar es el número de capas, número de 'neuronas', optimizador y el número de steps (equivalente a épocas), y como hiperparámetros especiales se tiene el grado de los splines y los intervalos de cuadrícula. Así que se usaron 8 neuronas en la entrada, con tres capas intermedias, la primera tiene 16 neuronas, la segunda tiene 12 neuronas y la tercera con 8 neuronas, finalizando la capa de salida con 4 neuronas correspondiente a 4 características predichas. Es importante ver que tienen los mismos

números de neuronas en la primera y última capa de ANN. EL optimizador fue 'LBFGS' con 30 steps. El número de steps se encontró con prueba y error, la función de pérdida no decae, sino que tiene una subida después de ciertos pasos, se halló de manera experimental que 30 es el número de steps más óptimo para este caso. Los hiperparámetros especiales fueron usados a como lo marca el repositorio de Liu et al. 2024 con Grid interval igual a 5, pero se modificó el grado de los splines, se usó cúbico y cuadrático.

### Preparación de Datos

#### Modelos KAN y ANN

Para estos modelos los datos se tomaron en conjunto, es decir, se hizo el entrenamiento con la red de las 4 características, usando el día anterior  $(n-1)^{th}$  y el anterior del anterior  $(n-2)^{th}$  durante las fechas de '2014-01-01' a '2024-04-01'. No se tomaron los datos en crudo, hubo un preprocesamiento antes de que los datos entraran a cada red. Primero se insertaron valores faltantes usando retropropagación hacia adelante, luego se le aplica la función 'detrend' de python Virtanen et al. 2020, que es para eliminar la tendencia de una serie de datos. En el contexto de series temporales, eliminar la tendencia significa ajustar y restar una tendencia lineal (o polinómica) de los datos para resaltar las fluctuaciones y patrones más pequeños. Para eliminar la tendencia calcula una línea recta que mejor se ajusta a los datos. Esto se hace usando una técnica de regresión lineal, donde se encuentra la línea que minimiza la suma de los cuadrados de las diferencias verticales entre los puntos de datos originales y la línea ajustada.

Después de todo lo anterior se aplica un escalamiento de  $[0, 1]$ , ahora sí se le aplica los modelos que se describieron anteriormente para calcular su predicción. Los datos de entrenamiento y prueba se dividieron en 80% y 20% respectivamente.

### Modelo CNN

En este modelo se generaron imágenes sintéticas en histogramas 2D. El modelo toma una matriz de  $20 \times 20$  generada del histograma 2D y genera una fracción como salida. La fracción predicha multiplicada por el día anterior  $(n-1)^{th}$  da el

valor de los datos predichos del  $n^{th}$  día. Un punto en el plano 2D representa el índice de los dos días consecutivos, donde la coordenada en el eje X es el precio del índice del día uno y la coordenada en el eje Y es el precio del índice del día dos. 50 conjuntos de puntos son considerados para generar un histograma 2D. Los histogramas que leen la red convolucional son histogramas en escalas de grises, cada histograma tiene 20 bins sobre los dos ejes, así se forman las matrices de  $20 \times 20$  y cada una de estas matrices alimenta la red convolucional para predecir el  $n^{th}$  día que corresponda en cada característica.

En Mukherjee et al. 2021 usan hiperparámetros muy pequeños, en este trabajo se usarán los siguientes, el optimizador será 'Adam', la función de costo es Mean Squared Error con una tasa de aprendizaje de 0.01, pero con 180 épocas esto para que diera un mejor resultado en las métricas. A los datos se le hace escalamiento antes de volverlos histogramas 2D, no se le aplica 'detrend' tiene un peor rendimiento la CNN.

## Resultados

Los resultados fueron increíblemente buenos usando KAN para splines cuadráticos, para splines cúbicos fue más o menos bueno. Por otro lado, el resultado con ANN no fue tan increíble como lo muestra el artículo Mukherjee et al. 2021, al parecer seguían una tendencia y por ende se obtenía el sobreajuste. Esto con el modelo de dicho artículo, pero se puede conseguir mejores resultados haciendo una optimización a ese modelo, incluso se puede tener mejores resultados comparados a KAN. Para CNN no se sabe si usaron otra cosa pues la arquitectura de la convolucional no estaba descrita al 100%. A continuación se deja una tabla con los resultados de cada modelo donde se midieron las métricas de 'Mean Squared Error', 'Mean Absolute Error' y ' $R^2$ ' del conjunto de prueba, para no poder los resultados de cada característica y tener 4 tablas o una tabla muy grande, se usará el promedio de cada métrica en conjunto de las características junto con un error que es la desviación estándar del conjunto.

Table 1. Resultados de los modelos en test para AMNZ

Métrica	ANN	CNN	KAN $k = 3$	KAN $k = 2$
$R^2$	0.4201	0.3511	0.7176	0.8604
MSE	0.0166	0.0129	0.0081	0.0040
MAE	0.0912	0.0920	0.0657	0.0482

Table 2. Error de los resultados de los modelos en test para AMZN

Métrica	ANN	CNN	KAN $k = 3$	KAN $k = 2$
$R^2$	$\pm 0.0113$	$\pm 0.1799$	$\pm 0.0399$	$\pm 0.0374$
MSE	$\pm 0.0004$	$\pm 0.0035$	$\pm 0.0012$	$\pm 0.0010$
MAE	$\pm 0.0017$	$\pm 0.0108$	$\pm 0.0038$	$\pm 0.0057$

Lo novedoso de usar la paquetería de Liu et al. 2024 es que te puede graficar el modelo y ver qué pesos se desconectaron. Si tiene pocas capas y pesos se podría observar qué grado de spline se ajustó a cada peso.

Table 3. Resultados de los modelos en test para GOOG

Métrica	ANN	CNN	KAN $k = 3$	KAN $k = 2$
$R^2$	0.8541	0.6532	0.9477	0.9489
MSE	0.0053	0.0064	0.0019	0.0019
MAE	0.0631	0.0652	0.0325	0.0322

Table 4. Error de los resultados de los modelos en test para GOOG

Métrica	ANN	CNN	KAN $k = 3$	KAN $k = 2$
$R^2$	$\pm 0.0107$	$\pm 0.0463$	$\pm 0.0065$	$\pm 0.0064$
MSE	$\pm 0.0003$	$\pm 0.0009$	$\pm 0.0002$	$\pm 0.0002$
MAE	$\pm 0.0019$	$\pm 0.0063$	$\pm 0.0022$	$\pm 0.0023$

Table 5. Resultados de los modelos en test para AAPL

Métrica	ANN	CNN	KAN $k = 3$	KAN $k = 2$
$R^2$	0.8761	0.2808	0.9287	0.9169
MSE	0.0026	0.0076	0.0015	0.0018
MAE	0.0397	0.0734	0.0303	0.0321

Table 6. Error de los resultados de los modelos en test para AAPL

Métrica	ANN	CNN	KAN $k = 3$	KAN $k = 2$
$R^2$	$\pm 0.0099$	$\pm 0.0769$	$\pm 0.0101$	$\pm 0.0101$
MSE	$\pm 0.0002$	$\pm 0.0005$	$\pm 0.0002$	$\pm 0.0002$
MAE	$\pm 0.0015$	$\pm 0.0032$	$\pm 0.0024$	$\pm 0.0021$

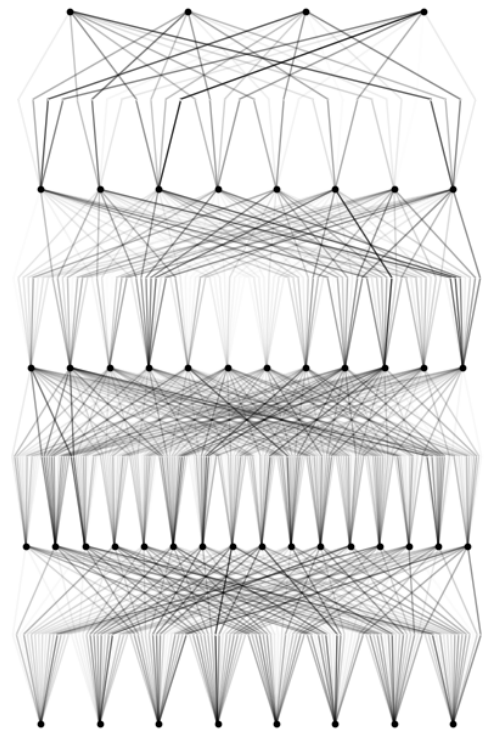


Figure 3. Modelo para la bolsa de Amazon usando  $k = 3$

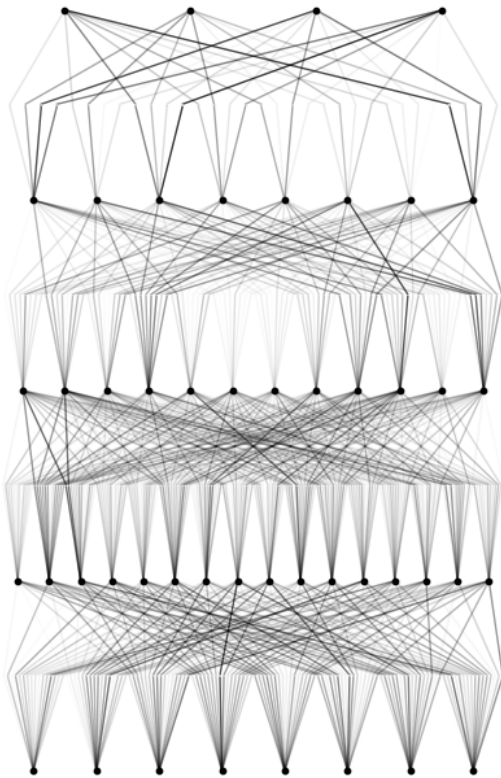


Figure 4. Modelo para la bolsa de Amazon usando  $k = 2$

### Conclusión

Con respecto a comprobar los resultados con Mukherjee et al. 2021, se aprecia que su modelo y preprocesamiento daban lugar a que sus resultados siguieran una tendencia y por ende estuvieran sobre ajustados. Incluso, viendo sus resultados se nota que los modelos no estaban hechos para predecir, sino para sobreajustar los datos ya existentes.

El nuevo modelo implementado dio mejores resultados que la ANN definida en Mukherjee et al. 2021, tal vez una ANN optimizada podría competir con KAN. Para trabajo futuro se propone hacer una ANN optimizada y compararla con la mejor KAN. De KAN se obtuvieron buenos resultados tanto con  $k = 2$  y  $k = 3$ , esto a pesar de eliminar la tendencia, lo que quiere decir que KAN es una gran propuesta y posiblemente un método en IA que pueda ser novedoso y potente.

Nota: A lo largo de este trabajo se hizo uso de OpenAI 2024 el cual se apoyó en crear diagramas, consultas de información rápidas y mejora de las gráficas proporcionadas a lo largo de este trabajo.

### References

Coqueret, Guillaume. 2021. Machine learning in finance: from theory to practice. *Quantitative Finance* 21 (1): 9–10. <https://doi.org/10.1080/14697688.2020.1828609>.

Liu, Ziming, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halver-son, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. 2024. *Kan: kolmogorov-arnold networks*. arXiv: 2404.19756 [cs.LG].

Mokhtari, Sohrab, Kang K. Yen, and Jin Liu. 2021. Effectiveness of artificial intelligence in stock market prediction based on machine learning. *International Journal of Computer Applications* 183, no. 7 (June): 1–8. issn: 0975-8887. <https://doi.org/10.5120/ijca2021921347>. <http://dx.doi.org/10.5120/ijca2021921347>.

Mukherjee, Somenath, Bikash Sadhukhan, Nairita Sarkar, Debajyoti Roy, and Soumil De. 2021. Stock market prediction using deep learning algorithms. *CAAI Transactions on Intelligence Technology* 8 (August). <https://doi.org/10.1049/cit2.12059>.

OpenAI. 2024. *Chatgpt: A conversational ai model*. <https://www.openai.com/chatgpt>. Accessed: 2024-06-06.

Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, et al. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17:261–272. <https://doi.org/10.1038/s41592-019-0686-2>.

Yahoo Finance. *Yahoo finance historical data*. <https://finance.yahoo.com/>. Accessed: 2024-06-04.

### Apéndice: Figuras

Se incluye una figura de cómo quedó la serie de tiempo usando KAN en el stock de AMZN con  $k = 2$  y  $k = 3$ :

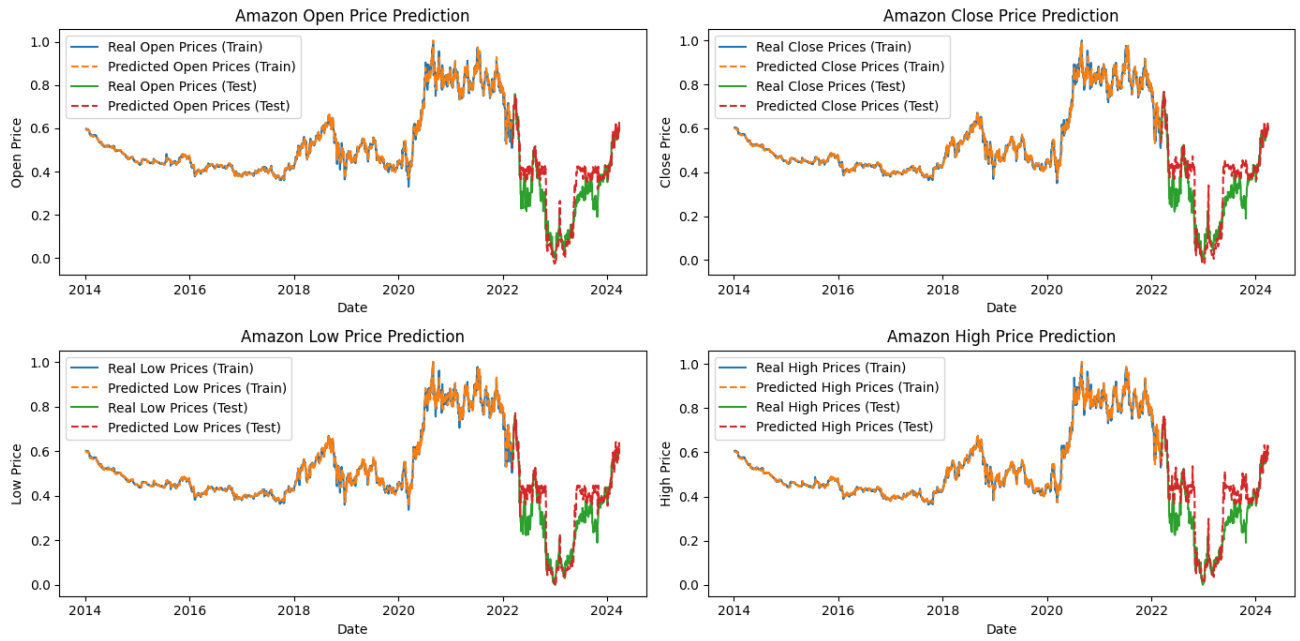


Figure 5. Serie de tiempo predicha contra la real usando KAN con  $k=3$

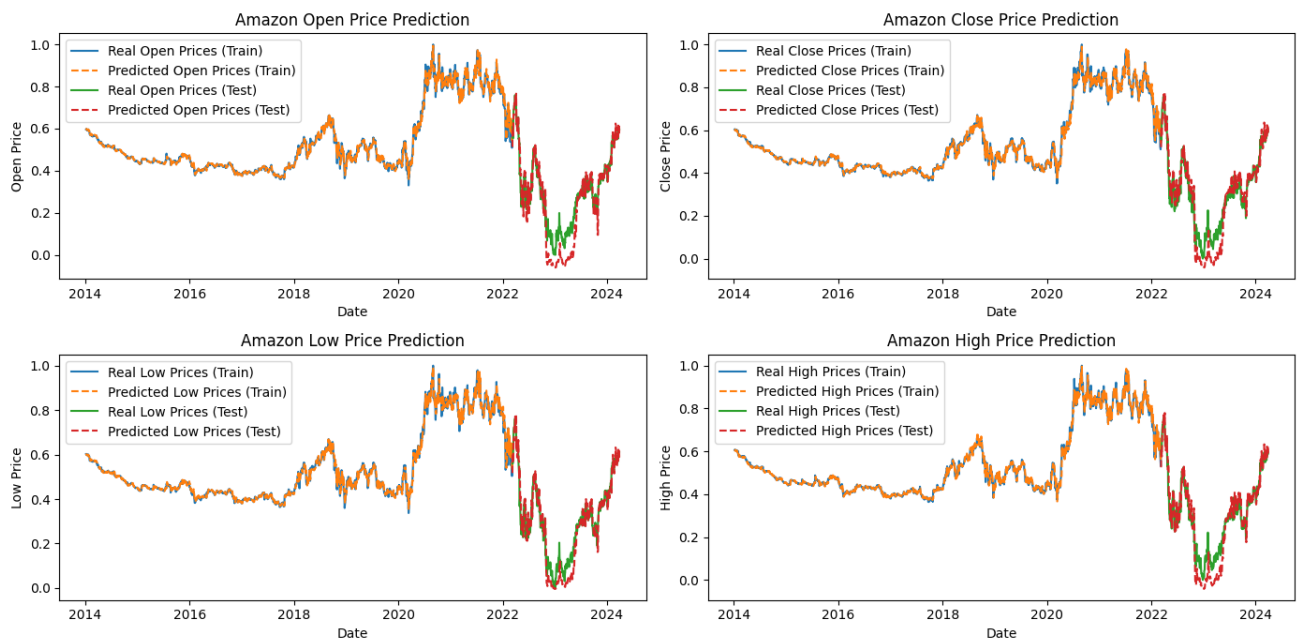


Figure 6. Serie de tiempo predicha contra la real usando KAN con  $k=2$