

# Aplicación de autoencoders y autoencoders variacionales al *sparse index tracking* del S&P100

Application of Autoencoders and Variational Autoencoders to Sparse Index Tracking of the S&P100

Daniel Aragón Urrego\*

---

\* Magíster en Finanzas, estudiante de doctorado en Ingeniería de Sistemas y Computación. Docente-investigador, Observatorio de Economía y Operaciones Numéricas (ODEON), Universidad Externado de Colombia, Bogotá (Colombia). [daniel.aragon@uexternado.edu.co]. [https://orcid.org/0000-0001-7677-1379].

Artículo recibido: 10 de diciembre de 2024

Aceptado: 15 de enero de 2025

Para citar este artículo:

Aragón-Urrego, D. (2025). Aplicación de autoencoders y autoencoders variacionales al *sparse index tracking* del S&P100. Odeon, 28, 87-102.

DOI: <https://doi.org/10.18601/17941113.n28.05>

## Resumen

Este artículo explora el uso de autoencoders (AE) y autoencoders variacionales (VAE) para abordar el problema de *sparse index tracking* aplicado al índice S&P100. Utilizando datos diarios de 2019 a 2023, se construyeron portafolios de seguimiento disperso con 5, 10, 15 y 20 activos, seleccionados mediante un enfoque de información comunal. Los resultados muestran que los VAE superan a los AE en términos de precisión y generalización, logrando un menor *empirical tracking error* en todos los portafolios. Este estudio destaca el potencial de los VAE como herramientas efectivas para replicar índices financieros bajo restricciones de cardinalidad, aunque no se consideraron costos de transacción ni otras fricciones del mercado.

**Palabras clave:** optimización de portafolios; aprendizaje profundo.

**Códigos JEL:** C63, G11

## Abstract

This article explores the use of autoencoders (AE) and variational autoencoders (VAE) to address the sparse index tracking problem applied to the S&P100 index. Using daily data from 2019 to 2023, sparse tracking portfolios with 5, 10, 15, and 20 assets were constructed, selected through a communal information approach. The results show that VAEs outperform AEs in terms of accuracy and generalization, achieving lower empirical tracking error across all portfolios. This study highlights the potential of VAEs as effective tools for replicating financial indices under cardinality constraints, although transaction costs and other market frictions were not considered.

**Keywords:** Portfolio optimization; deep learning.

**JEL codes:** C63, G11

## Introducción

En la gestión de portafolios, las estrategias de inversión se clasifican en activas y pasivas. Las estrategias activas buscan aprovechar las ineficiencias del mercado para generar rendimientos superiores al índice de referencia, pero suelen implicar costos operativos elevados debido a la frecuencia de transacciones y la necesidad de análisis constante (Fama, 1970). Por el contrario, las estrategias pasivas, basadas en la hipótesis del mercado eficiente (EMH), reconocen que superar consistentemente al mercado es improbable. En este contexto, el *index*

*tracking* se posiciona como una estrategia pasiva que busca replicar el rendimiento de un índice de referencia, como el S&P100 o el Nasdaq 100.

El *index tracking* (IT) ofrece ventajas significativas, como menores costos operativos y simplicidad en la gestión, pero presenta limitaciones, como la exposición completa a los riesgos del índice y la incapacidad de ajustarse a condiciones de mercado desfavorables (Bogle, 1999). Además, el enfoque tradicional de replicar completamente el índice (*full index tracking*), conlleva desafíos como altos costos de transacción y la inclusión de activos ilíquidos. En contraste, el *sparse index tracking* se propone como una solución más eficiente, seleccionando un subconjunto representativo de activos para minimizar los costos sin comprometer la representatividad del portafolio.

Este artículo aborda el problema de *sparse index tracking* desde una perspectiva innovadora, empleando modelos de aprendizaje profundo como los AE y VAE. Estos modelos no solo reducen dimensionalidad, sino que también capturan relaciones complejas entre los activos, lo que permite construir portafolios representativos con bajos niveles de *tracking error*. A lo largo del estudio, se evalúa el desempeño de ambos modelos utilizando datos del S&P100 entre 2019 y 2023, destacando las ventajas del enfoque probabilístico de los VAE en la construcción de portafolios eficientes bajo restricciones de cardinalidad.

## 1. Index Tracking

En la gestión de portafolios, las estrategias de inversión se pueden clasificar en dos categorías principales: activas y pasivas, dependiendo del enfoque adoptado por el gestor. Las estrategias activas se basan en la premisa de que los mercados financieros presentan ineficiencias que pueden ser aprovechadas para generar rendimientos superiores a los del índice de referencia. Este enfoque implica un análisis constante del mercado, la identificación de oportunidades de inversión y la realización de ajustes dinámicos en el portafolio. Sin embargo, este tipo de gestión a menudo conlleva costos de transacción más altos debido a la frecuencia de las operaciones y a la necesidad de investigación continua (Fama, 1970).

Por otro lado, las estrategias pasivas reconocen que, debido a la eficiencia de los mercados, superar consistentemente al índice es improbable. Este concepto se basa en la hipótesis del mercado eficiente (EMH), que sugiere que toda la información disponible ya está reflejada en los precios de los activos, lo que hace difícil para los inversores obtener rendimientos superiores de manera sistemática.

(Fama, 1970). En lugar de intentar superar al mercado, las estrategias pasivas buscan replicar el comportamiento del índice de referencia.

Dentro de las estrategias pasivas se encuentra el *index tracking* (IT), o seguimiento de índices, que consiste en construir un portafolio que reproduzca el rendimiento de un índice específico, como el S&P 500, el Nasdaq 100, o cualquier otro *benchmark*. Esta estrategia se implementa a menudo a través de fondos indexados o ETF (fondos cotizados en bolsa), que están diseñados para seguir de cerca el rendimiento de un índice determinado.

El *index tracking* ofrece varias ventajas. En primer lugar, tiende a tener costos más bajos en comparación con las estrategias activas, ya que requiere menos transacciones y menos investigación. Además, los fondos indexados han demostrado, en muchos casos, que superan a la mayoría de los fondos de gestión activa en términos de rendimiento neto a largo plazo, especialmente después de considerar las comisiones y los costos (Malkiel, 2003).

Sin embargo, el *index tracking* también tiene sus desventajas. Al replicar un índice, los inversores están expuestos a todos los riesgos asociados con ese índice, lo que significa que no hay protección contra caídas del mercado. Además, en mercados en declive, los fondos indexados pueden experimentar pérdidas significativas, ya que no tienen la flexibilidad de ajustar las posiciones para mitigar el riesgo (Bogle, 1999).

### 1.1 Full index tracking vs. sparse index tracking

El *index tracking* puede abordarse desde dos perspectivas principales:

**Full index tracking:** en este enfoque, el portafolio incluye todos los activos que componen el índice en las mismas proporciones. Por ejemplo, para replicar el S&P500, se tomaría posición en las 500 principales acciones estadounidenses. Aunque este enfoque garantiza una alta precisión en la replicación del índice, presenta importantes desventajas (Strub y Baumann 2018; Benidis *et al.*, 2018):

- Altos costos de transacción: mantener posiciones en un gran número de activos genera costos considerables, especialmente en índices amplios.
- Rebalanceos frecuentes: los índices suelen actualizar sus componentes, lo que obliga a realizar ajustes continuos en el portafolio.
- Activos ilíquidos: algunos componentes del índice pueden tener poca liquidez, dificultando su inclusión en el portafolio.

**Sparse index tracking:** este enfoque busca replicar el índice seleccionando solo un subconjunto de activos representativos. Al reducir el número de componentes, se minimizan los costos operativos y se simplifica la gestión. Sin

embargo, este enfoque plantea el desafío de garantizar que el portafolio seleccionado mantenga una alta representatividad del índice.

## 1.2 Tracking error

El éxito del *index tracking* se mide mediante el *tracking error* (TE), una métrica que cuantifica la desviación entre los retornos del portafolio y los del índice. Matemáticamente, el *tracking error* se define como:

$$TE(w) = \sqrt{\frac{1}{T} \sum_{t=1}^T (r_t - X_t w)^2} \quad (1)$$

Donde:

T: número de periodos u observaciones, por ejemplo, días o meses.

$r_t$ : retorno del índice de referencia (*benchmark*) en cada periodo  $t$ .

$X_t$ : matriz de retornos de los activos en cada período  $t$ .

$w$ : vector de pesos de cada activo dentro portafolio. Generalmente, se asume que los pesos deben cumplir las siguientes restricciones:

$w_i \geq 0$ : restricción para garantizar únicamente posiciones largas.

$\sum_{i=1}^n w_i = 1$ : restricción de asignación total de recursos.

Así, un *tracking error* bajo indica que el portafolio está siguiendo de cerca al índice, mientras que un *tracking error* alto sugiere que hay una mayor desviación entre los retornos del portafolio y los del índice.

Ahora bien, minimizar el *tracking error* implica encontrar un portafolio cuyas dinámicas de retorno sigan de cerca las del índice. Sin embargo, cuando se introducen restricciones de cardinalidad, el problema se vuelve no convexo, aumentando su complejidad computacional (Shu *et al.*, 2020; Coleman *et al.*, 2006).

## 1.3 Enfoques para resolver el problema

En la literatura sobre *index tracking* se han identificado varios enfoques para abordar este problema. En este sentido, Silva y de Almeida (2024) ofrecen una revisión de los trabajos realizados entre 2010 y 2021, identificando tres categorías principales que han guiado la investigación en este campo: programación matemática, técnicas estadísticas y otros enfoques:

- Programación matemática: métodos como la programación cuadrática mixta (MIQP) y la programación entera mixta (MIP) formulan el problema de *sparse index tracking* de manera explícita, incluyendo restricciones de cardinalidad. Un ejemplo destacado es el trabajo de Mutunge y Haugland (2018), quienes implementaron MIQP para minimizar el *tracking error* bajo restricciones de cardinalidad. Además, demostraron que la minimización del *tracking error* sujeto a restricciones de cardinalidad es un problema *NP-hard*.
- Técnicas estadísticas: estas técnicas utilizan modelos de regresión y regularización para seleccionar activos y asignar pesos. Aquí, uno de los trabajos más conocidos lo desarrollaron Benidis *et al.* (2017) quienes aplicaron regularización para construir portafolios dispersos en entornos de alta dimensionalidad, destacando la efectividad de estas técnicas para reducir el *tracking error*.
- Otros enfoques: en esta categoría se destacan los métodos de aprendizaje automático, que aprovechan la capacidad de las redes neuronales para modelar relaciones complejas entre los activos del índice. Entre los trabajos que han implementado este enfoque se encuentran: Ouyang *et al.* (2019), Ni y Zhang (2013), Shu *et al.* (2020), Zhang *et al.* (2020), Kim y Kim (2019).

#### 1.4 Autoencoders

Los autoencoders (AE), introducidos por Geoffrey Hinton y Ruslan Salakhutdinov en 2006, son un tipo de red neuronal diseñada para la reducción de dimensionalidad y la extracción de características latentes. Estas redes aprenden a generar una representación comprimida de los datos de entrada, permitiendo reconstruirlos con alta precisión. Al obligar a la red a sintetizar la información en un espacio más compacto, los autoencoders logran identificar las características más relevantes de los datos, eliminando ruido y redundancias.

En su trabajo seminal, Hinton y Salakhutdinov demostraron la eficacia de los autoencoders en tareas como la reducción de dimensionalidad y el preentrenamiento de redes neuronales profundas. Este enfoque revolucionó la forma en que se aborda el aprendizaje en modelos complejos, destacando su potencial en una amplia gama de aplicaciones (Hinton y Salakhutdinov, 2006), siendo una de las más comunes la compresión de imágenes. Al aprender a representar imágenes de alta dimensión en un espacio de menor dimensión, los autoencoders pueden reducir significativamente el tamaño de los archivos de imagen sin perder información crítica. Esto es especialmente útil en contextos donde el almacenamiento y la transmisión de datos son limitados (Vincent *et al.*, 2010).

Además, los autoencoders son efectivos en la eliminación de ruido, un proceso conocido como *denoising*. En este enfoque, se entrena al autoencoder para reconstruir datos originales a partir de versiones ruidosas de esos datos, lo que permite mejorar la calidad de las señales y las imágenes (Vincent *et al.*, 2008).

Más recientemente, los autoencoders han encontrado aplicaciones en el ámbito financiero, particularmente en la construcción de portafolios. Al utilizar autoencoders para analizar grandes volúmenes de datos financieros, se pueden identificar patrones y relaciones ocultas que pueden no ser evidentes a través de métodos tradicionales. Esto permite una mejor toma de decisiones en la asignación de activos y la gestión de riesgos (Bontempi *et al.*, 2013).

Un AE consta de dos componentes principales:

- Codificador (*encoder*): reduce la dimensionalidad de los datos de entrada ( $x$ ), mapeándolos a un espacio latente ( $z$ ) más compacto. Matemáticamente, se representa como:

$$z = f(W_e x + b_e) \quad (2)$$

Donde:

$W_e$  y  $b_e$ : vector de pesos y sesgo en el codificador, respectivamente

$f(\cdot)$ : función de activación, generalmente ReLU.

- Decodificador (*decoder*): reconstruye los datos originales desde la representación latente ( $z$ ), generando una salida ( $x'$ ).

$$x' = g(W_d z + b_d) \quad (3)$$

Donde:

$W_d$ : representa los pesos

$b_d$ : sesgos en el decodificador

$f(\cdot)$ : función de activación.

El objetivo de los autoencoders es minimizar el error de reconstrucción entre los datos originales ( $x$ ) y la reconstrucción ( $x'$ ), utilizando una función de pérdida, generalmente la norma  $\ell_2$ :

$$L(x, x') = \|x - x'\|^2 \quad (4)$$

En el contexto del *sparse index tracking*, los autoencoders podrían utilizar los retornos de los activos como entrada para generar una representación latente compacta que capture las relaciones más relevantes entre estos. A partir de esta representación, se identifican las dimensiones clave que permiten seleccionar un subconjunto reducido de activos representativos del índice. Luego, el decodificador reconstruye los retornos del índice utilizando únicamente este subconjunto, asegurando que el error de reconstrucción y, por ende, el *tracking error*, se mantenga bajo. Finalmente, se optimizan los pesos asignados a los activos seleccionados para minimizar la desviación con respecto al índice, respetando las restricciones de cardinalidad. Este proceso combina reducción de dimensionalidad, selección de activos y optimización en un solo marco eficiente y escalable.

Sin embargo, Zhang *et al.* (2020) señalaron que los autoencoders tienen ciertas limitaciones relacionadas con la pérdida de información cuando se comprimen los datos en una representación latente excesivamente reducida, lo que puede comprometer la calidad de la reconstrucción. Además, los autoencoders tradicionales no modelan explícitamente la incertidumbre en el espacio latente, lo que puede dificultar su capacidad de generalización a nuevos datos. Estas observaciones resaltan la necesidad de utilizar autoencoders variacionales (VAE), que integran un enfoque probabilístico para capturar mejor la distribución subyacente de los datos y mitigar la pérdida de información, logrando así una mayor robustez y precisión en aplicaciones como el *sparse index tracking*.

### 1.5 Autoencoders variacionales

Los autoencoders variacionales (VAE, por sus siglas en inglés) fueron propuestos por Kingma y Welling (2019) y son una extensión de los autoencoders tradicionales al incorporar un enfoque probabilístico en la representación de datos. En lugar de mapear directamente los datos de entrada ( $x$ ) a un punto específico en el espacio latente ( $z$ ), los VAE modelan una distribución condicional  $q(z|x)$ , generalmente una distribución normal multivariada. Esto permite que la representación latente capture incertidumbre y variabilidad en los datos.

La arquitectura de un VAE incluye un codificador que genera los parámetros de la distribución  $q(z|x)$ , y un decodificador que reconstruye ( $x$ ) a partir de muestras generadas en el espacio latente. Este enfoque no solo garantiza continuidad en el espacio latente, sino que también habilita el uso de técnicas de optimización modernas, como la retropropagación, gracias al truco de reparametrización:



$$z = \mu + \sigma^2 \epsilon, \text{ en donde } \epsilon \sim \mathcal{N}(0,1) \quad (5)$$

Este proceso asegura que el modelo pueda entrenarse de manera eficiente, incluso al trabajar con distribuciones probabilísticas.

Ahora bien, en términos de la función de pérdida, el entrenamiento de un VAE se basa en maximizar el límite inferior de evidencia (*Evidence Lower Bound*, ELBO), que equilibra dos objetivos fundamentales:

- Error de reconstrucción: busca garantizar que los datos originales  $x$  puedan ser reconstruidos desde la representación latente:

$$\mathbb{E}_{q(z|x)}[\log p_0(x|z)] \quad (6)$$

- Regularización del espacio latente: implica alinear la distribución posterior  $q(z|x)$  con una distribución previa  $p(z)$ , generalmente una normal estándar  $\mathcal{N}(0,1)$ , mediante la divergencia Kullback Leibler (KL):

$$D_{KL}(q(z|x)||p(z)) \quad (7)$$

Con lo cual, la función de pérdida del VAE consiste en:

$$Loss = -ELBO = L(x, x') + \lambda \sum_i D_{KL}(q(z_i|x)||p(z_i)) \quad (8)$$

En donde  $L(x, x')$  penaliza los errores de reconstrucción, y  $\lambda$  regula el balance entre la precisión en la reconstrucción y la regularización del espacio latente.

Considerando el problema del *sparse index tracking*, los VAE pueden ser útiles para modelar los retornos de los activos  $X$  en un marco probabilístico, permitiendo seleccionar un subconjunto reducido de activos que mantienen una alta representatividad del índice, minimizando el *tracking error*.

## 2. Aplicación

### 2.1 Datos

Para la aplicación numérica de los modelos de *sparse index tracking*, se utilizó como índice de referencia el S&P100. Se identificaron 99 activos constituyentes

del índice que cuentan con información completa para el periodo de análisis comprendido entre enero de 2019 y diciembre de 2023. Dos activos fueron excluidos debido a datos faltantes. Se emplearon precios de cierre ajustados diarios para los activos y el índice, obtenidos a través de la API de Yahoo Finance, lo que resultó en un total de 1258 observaciones por cada variable. Así mismo, no se consideran fricciones de mercado como costos de transacción, impuestos, entre otros.

## 2.2 Metodología

Se implementaron algoritmos basados en autoencoders (AE) y autoencoders variacionales (VAE), diseñados para reducir dimensionalidad y seleccionar subconjuntos representativos de activos que minimicen el *tracking error*. Para medir la capacidad de los modelos, se construyeron portafolios de seguimiento dispersos con 5, 10, 15 y 20 activos, seleccionados utilizando un enfoque de información comunal que consiste en utilizar los algoritmos AE y VAE para medir el grado de información compartida entre el índice y sus activos constituyentes. Siguiendo las propuestas de Heaton *et al.* (2017) y Zhang *et al.* (2020), se calculó la similitud entre los datos originales ( $x$ ) y los datos reconstruidos ( $x'$ ) mediante la siguiente métrica:

$$d_i = \|x - x'\|^2 \quad (9)$$

En donde un valor más bajo de  $d_i$  indica una mayor cantidad de información comunal entre el activo y el portafolio de mercado. Los activos seleccionados para formar los portafolios de tamaño  $k$ , con  $k < n$ , fueron aquellos que presentaron los valores más bajos de  $d_i$ , asegurando que el subconjunto elegido fuese representativo de las dinámicas del índice.

Para la implementación y el entrenamiento de los modelos AE y VAE, se utilizó Python, empleando las bibliotecas TensorFlow y Keras debido a su flexibilidad y eficiencia en el manejo de redes neuronales profundas. Se llevaron a cabo dos configuraciones experimentales distintas:

- a) Periodos fijos de entrenamiento y prueba. El conjunto de datos se dividió en dos periodos:

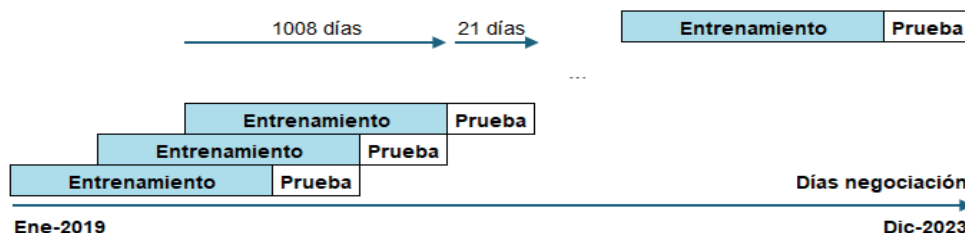
Entrenamiento: de enero de 2019 a diciembre de 2022 (4 años, 1008 días).

Prueba: todo el año 2023 (250 días).

En esta configuración, los modelos fueron entrenados con los datos del periodo de entrenamiento y evaluados con los datos del periodo de prueba para medir su desempeño *out-of-sample*.

- b) Método de ventanas deslizantes. Se aplicó un enfoque de ventanas deslizantes, utilizando 4 años de datos de entrenamiento (1008 días) y 1 mes de datos de prueba (21 días) en cada iteración. Este método permite capturar la evolución temporal del mercado y adaptar continuamente los modelos a nuevas condiciones. Los periodos de prueba abarcaron de enero a diciembre de 2023, con un rebalanceo mensual del portafolio, como se ilustra en la figura 1.

Figura 1. Esquema de ventanas deslizantes



Fuente: elaboración propia.

La optimización de los modelos AE y VAE incluyó la selección de hiperparámetros tales como el número de capas, neuronas por capa, funciones de activación (ReLU), optimizadores y técnicas de regularización como L2. Para minimizar la función de pérdida, se utilizó el optimizador Adam con una tasa de aprendizaje de  $10^{-3}$ . Así mismo, cada red neuronal fue entrenada durante 150 épocas, utilizando un tamaño de lote (*batch size*) de 15.

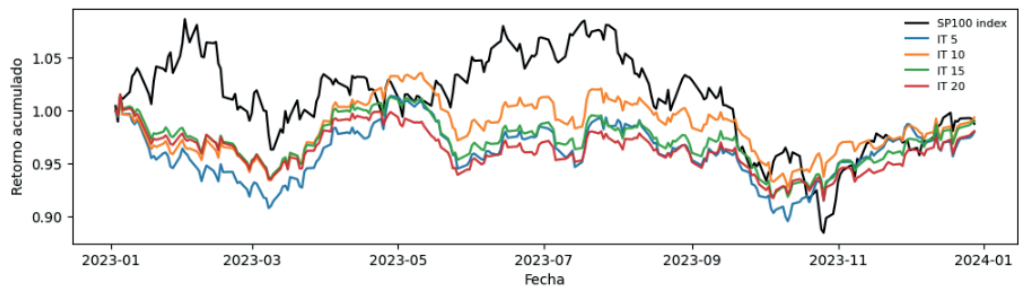
## 2.3 Resultados

La presentación de los resultados comienza con una comparación entre los modelos propuestos: autoencoder (AE) y autoencoder variacional (VAE). En este contexto, la figura 2 ilustra el retorno acumulado tanto del índice de referencia como de los cuatro portafolios de seguimiento construidos utilizando los modelos AE y VAE, considerando cardinalidades de 5, 10, 15 y 20 activos.

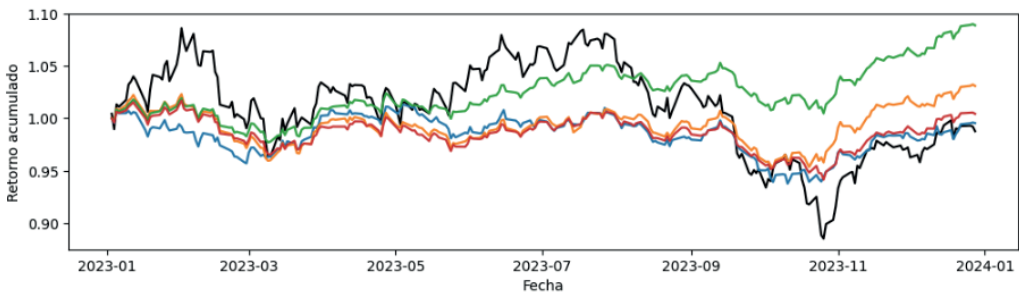
Es importante destacar que, en términos de retorno acumulado, los portafolios generados con el modelo AE muestran un seguimiento más cercano al índice. En

el caso del modelo VAE, el portafolio que incluye 5 activos presenta el retorno acumulado más cercano al índice, lo que indica una buena capacidad de replicación a pesar de su menor cardinalidad. Sin embargo, los portafolios de 10, 15 y 20 activos del modelo VAE terminan con retornos acumulados mayores. Cabe señalar que este análisis se ha realizado sin considerar los costos de transacción, lo que permite una evaluación más clara de la efectividad de los modelos en la replicación del rendimiento del índice.

Figura 2. Modelos AE y VAE, retorno acumulado de índice y portafolios de seguimiento  
(2a) Portafolios de seguimiento, modelo AE



(2b) Portafolios de seguimiento, modelo VAE



Fuente: elaboración propia.

La tabla 1 presenta el *empirical tracking error* para cada uno de los cuatro portafolios considerados en el periodo *out-of-sample* (año 2023). En general, se observa que el modelo VAE entrega mejores resultados que el AE para cada uno de los portafolios de seguimiento. Por ejemplo, en promedio, el modelo AE presenta un ETE del 0,011 frente al 0,004 del modelo VAE. Este comportamiento

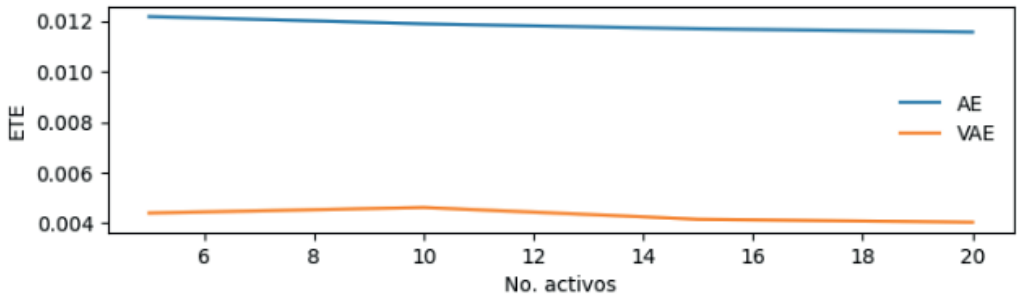
también se aprecia en la figura 3, en donde, conforme se incrementa la cantidad de activos en el portafolio, pasando de 5 hasta 20, el ETE disminuye, siendo menor el valor en el caso del VAE.

Tabla 1. Comparación de empirical tracking error (ETE) para diferentes portafolios

Modelo	No. Activos			
	5	10	15	20
ETE _AE	0,012174	0,011892	0,011892	0,011563
ETE _VAE	0,004400	0,004624	0,004153	0,004040

Fuente: elaboración propia.

Figura 3. Modelos AE y VAE, evolución del empirical tracking error (ETE) para diferentes cantidades de activos

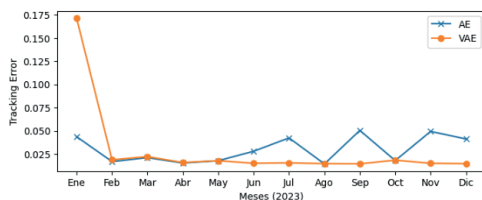


Fuente: elaboración propia.

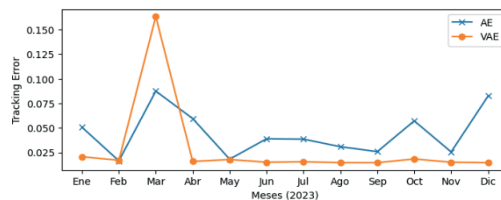
En la figura 4 se observa la evolución del ETE en los modelos de AE y VAE para los cuatro portafolios en el periodo fuera de muestra (2023), considerando una arquitectura con cinco dimensiones latentes. Se destacan dos elementos con respecto al ETE: i) La métrica disminuye conforme se incrementa la cantidad de activos en el portafolio de seguimiento, lo cual era de esperarse; ii) los resultados para el VAE son menores que para el AE, lo cual puede atribuirse al enfoque probabilístico que le permite capturar de una mejor forma la incertidumbre y las relaciones no lineales entre los activos.

Figura 4. Modelos AE y VAE, empirical tracking error (ETE) por portafolio de seguimiento

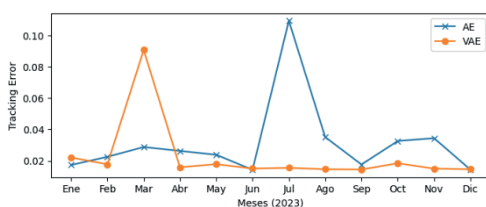
(4a) ETE para portafolio de 5 activos



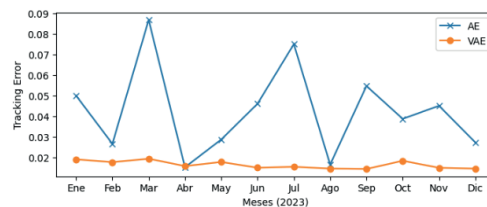
(4b) ETE para portafolio de 10 activos



(4c) ETE para portafolio de 15 activos



(4d) ETE para portafolio de 20 activos



Fuente: elaboración propia.

### 3. Conclusiones

La replicación del desempeño del índice S&P100 utilizando autoencoders (AE) y autoencoders variacionales (VAE) se presenta como una estrategia innovadora en la construcción de portafolios *sparse*. Así, mediante estas técnicas se selecciona un subconjunto reducido de activos que reflejan el comportamiento del índice, optimizando recursos y reduciendo costos asociados con la diversificación completa. En particular, los VAE destacan por su enfoque probabilístico, que captura de manera más precisa la incertidumbre inherente al mercado y las relaciones no lineales entre los activos, logrando un menor *tracking error* en comparación con los AE tradicionales. Esta capacidad los convierte en una herramienta poderosa para gestionar portafolios pasivos en finanzas.

Además de la ventaja que supone el modelado de relaciones complejas en el espacio latente, el uso de regularización ayuda a que los modelos mantengan un equilibrio entre precisión y generalización, evitando el sobreajuste.

A pesar de estos resultados prometedores, existen limitaciones que deben considerarse. El estudio no incorpora fricciones de mercado, como costos de transacción, impuestos u otras restricciones operativas, que son elementos clave en la implementación práctica de cualquier estrategia de inversión. Estas

fricciones pueden alterar de manera significativa el desempeño proyectado del portafolio, lo que reduce su viabilidad en contextos reales. Incluir estas consideraciones en futuras investigaciones permitirá evaluar de manera integral la aplicabilidad de estas técnicas y proponer mejoras que las hagan más robustas frente a los desafíos del mercado.

## Referencias

- Benidis, K., Feng, Y. y Palomar, D. (2018). Sparse portfolios for high-dimensional financial index tracking, *IEEE Trans. Signal Processing*, 66(1), 155-170. 10.1109/TSP.2017.2762286.
- Bogle, J. C. (1999). *Common sense on mutual funds: New imperatives for the intelligent investor*. John Wiley & Sons.
- Bontempi, G., Ben Taieb, S. y Lecue, F. (2013). Machine learning strategies for time series forecasting. *European Journal of Operational Research*, 234(2), 361-370.
- Coleman, T. F., Li, Y. y Henniger, J. (2006). Minimizing tracking error while restricting the number of assets. *Journal of Risk*, 8(4), 33.
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383-417.
- Feng, Y. y Palomar, D. P. (2016). A signal processing perspective on financial engineering. *Foundations and Trends® in Signal Processing*, 9(1-2), 1-231.
- Heaton, J. B., Polson, N. G. y Witte, J. H. (2017). Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1), 3-12. <https://doi.org/10.1002/asmb.2209>.
- Hinton, G. E. y Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks, *Science*, 313(5786), 504507. <https://doi.org/10.1126/science.1127647>
- Kim, S. y Kim, S. (2019). Index tracking through deep latent representation learning. *Quantitative Finance*, 20(4), 639-652. <https://doi.org/10.1080/14697688.2019.1683599>
- Kingma, D. P. y Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4), 307-392.
- Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 17(1), 59-82.
- Mutunge, P. y Haugland, D. (2018). Minimizing the tracking error of cardinality constrained portfolios. *Computers & Operations Research*, 90, 33-41. <https://doi.org/10.1016/j.cor.2017.09.002>
- Ni, L. y Zhang, J. (2013). Portfolio optimization for index investing based on self-organizing neural network. In KIm, Y. H. t Yarlagadda, P., editor, *Sensors, Measurement and intelligent Materials* (vol. 303-306, pp. 1595-1598). *Applied Mechanics and Materials*.

- Ouyang, H., Zhang, X., Yan, H. (2019). Index tracking based on deep neural network. *Cognitive Systems Research*, 57, 107-114. <https://doi.org/10.1016/j.cogsys.2018.10.022>
- Shu, L., Shi, F., Tian, G. (2020). High-dimensional index tracking based on the adaptive elastic net. *Quantitative Finance*, 20(9), 1513-1530. <https://doi.org/10.1080/14697688.2020.1737328>.
- Silva, J. C. S. y de Almeida Filho, A. T. (2024). A systematic literature review on solution approaches for the index tracking problem. *IMA Journal of Management Mathematics*, 35(2), 163-196.
- Strub, O. y Baumann, P. (2018). Optimal construction and rebalancing of index-tracking portfolios. *European Journal of Operational Research*, 264(1), 370-387.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. y Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8), 1439-1451.
- Vincent, P., Lajoie, I., Bengio, Y. y Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11, 3371-3408.
- Wang, Y. J., Wu, L. y Wu, L. (2024). An integrative extraction approach for index-tracking portfolio construction and forecasting under a deep learning framework. *The Journal of Supercomputing*, 80, 2047-2066. <https://doi.org/10.1007/s11227-023-05538-z>.
- Zhang, C., Liang, S., Lyu, F. y Fang, L. (2020). Stock-Index Tracking Optimization Using Auto-Encoders. *Frontiers in Physics*, 8, 388. <https://doi.org/10.3389/fphy.2020.00388>