Building Deep Tracking Portfolios through deep learning tools: one application case of Autoencoders and Variational Autoencoders

Daniel Aragón Urrego[1] , Oscar Eduardo Reyes Nieto[2] and Sarah Daniela Coral Zuñiga[3]

[1] Universidad Nacional, Bogotá, Colombia. [daragonu@unal.edu.co]
[2] Universidad Nacional, Bogotá, Colombia. [oreyesn@unal.edu.co]
[3] Universidad Nacional, Bogotá, Colombia. [scoralz@unal.edu.co]

Abstract. The reducing on number of assets in investment portfolios regarding one market index basket could lead toward important benefits for market agents like facing lower costs and upgrading portfolio management in general. This paper explores deep learning techniques to aim this objective. Thus, by autoencoders (AE) and variational autoencoders (VAE), cardinality constraints to limit the number of assets in a reference portfolio are incorporated in the model programming. A training period from 2019 to 2022 of historical returns were considered, while the year 2023 was reserved for performance validation. According to the results, tracking portfolios constructed by both AE and VAE closely track the index in both in-sample and out-of-sample periods with the reduced number of assets and highlighting the strengths and weaknesses of each approach. Other two approaches of regularization were considered to solve potential overfitting problems: dropout and L2 regularization. In this way, tracking portfolios were modeled regarding the US Nasdaq 100 market index (NDX).

Keywords: Index Tracking, Deep Learning, Autoencoders, Variational Autoencoders, Portfolio Theory.

1. Introduction.

Index Tracking strategies lead to replicate the performance of a market index (benchmark) with the aim of minimizing the tracking error measure (TE), and thus be able to generate cumulative returns as close as those ones of the benchmark [1] [2]. This implies the construction of a reduced portfolio by number of assets that being representative enough to replicate the benchmark. This approach has the potential advantages of reducing transaction costs, avoids holding very small or illiquid positions, and facilitates portfolio management [3].

The inclusion of a constraint on the amount of assets in the portfolio leads to a specific problem known as sparse index-tracking (IT), which implies cardinality constraints [4]. The number of assets bounded by the portfolio constraint results in an NP-hard optimization problem due to the non-differentiability and non-convexity of the search space [5] [6]. Besides this, the implementation could be computationally expensive, particularly in high-dimensional environments [3] [4].

Recent works has focused on implementing machine learning (ML) techniques to address the computational complexity of such problems. For instance, Zhang et al. [7] implemented a variational autoencoder (VAE) demonstrating that an AE with a deeper latent layer contains a single node that may result in an excessive loss of input information. To mitigate these issues, this paper extends the application of deep learning models, specifically autoencoders (AE) and variational autoencoders (VAE), for index tracking with cardinality constraints. The empirical study indicates that the proposed VAE algorithm achieves superior out-of-sample results compared to traditional methods.

Additional to this introduction, this paper describes the most recognized approaches of the index tracking problem in section 2. Numerical applications are presented in section 3. Finally, conclusions are exposed in section 4.

2. Index Tracking and Deep Neural Networks (DNN).

2.1. Index Tracking Model.

The Mean-Variance (MV) model considers n risky assets that are characterized by the random nature of their returns, and under the assumption that the returns follow a normal distribution. From the historical returns $E(r_i) = \mu$ and the respective covariance matrix (Σ), the expected returns and the variance of the portfolio are constructed considering the weight (w_i) of each risky asset i [8] [9].

$$E(r_p) = W'\mu \quad \text{--} \quad \sigma_p^2 = W'\,\Sigma\,W \qquad (1)$$

Then, the optimization problem consists of minimizing the portfolio variance (σ_p^2) subject to one budget constraint and one given expected return, where $\mathbf{1} \in \mathbb{R}^{n\times 1}$ is a vector of ones.

$$\min_{\{w\}} \{w'\Sigma w\} \quad \text{s.t.} \begin{cases} w'\mu = \mu_p \\ w'\mathbf{1} = 1 \end{cases} \tag{2}$$

According to the MV approach, passive strategies suggest that the best path to take for the portfolio manager is following the market given that the market cannot be overcome. In this sense, a strategy it is proposed to take a small amount of assets from the index basket through one cardinality constraint which one is known as the sparse approach. For its part, the empirical tracking error (ETE) is defined as a measure that allows to know how much the tracking portfolio replicates the benchmark from a small number of assets considered [3] [10] [11].

$$ETE(w) = \frac{1}{T}\|r_b - \mathrm{X\,W}\|_2^2 \tag{3}$$

where, X represents the returns of the N assets and, W represents the weights of the assets in the sparse portfolio, with $w_i \geq 0$ and $w'\mathbf{1} = 1$. On the other hand, r_b represents the returns of the index and T are the days. Now, it is proposed to unify the selection of assets and the allocation of weights by directly penalizing the cardinality of the tracking portfolio [3].

$$\min_{\{w\}} ETE\ (w) = \frac{1}{T}\|r_b - \mathrm{Xw}\|_2^2 + \lambda\|w\|_0 \quad \text{s.t.} \begin{cases} w'\mathbf{1} = 1 \\ w \geq 0 \end{cases} \tag{4}$$

where $\lambda \geq 0$ is a parameter that controls the sparsity of the portfolio. It implies that more dispersed solutions are obtained for higher values of λ. Thus, both the selection and the allocation of capital are carried out. This problem is highly non-convex due to the term $l_0 - norm$.

2.2. Approaches Applied to Index Tracking.

Research on index tracking has explored several methods over the past decade. Silva and Almeida [12] identified three main approaches: (i) Mathematical Programming, which one includes using mixed integer programming (MIP) to minimize tracking errors, such as the work by Mutunge and Haugland [13], who used mixed integer quadratic programming (MIQP); (ii) Statistical Techniques, involving regression with regularization and variable selection, as seen in studies by Benidis et al. [3] and Feng and Palomar [14], who addressed the problem through staged asset selection and capital allocation, or by penalizing the number of assets directly; and (iii) Machine Learning, such as neural networks, applied by Shu et al. [4], Zhang et al. [7], Ouyang et al. [15], among others.

It is precisely within this third category that machine learning (ML) algorithms have made significant advances in recent decades and have been widely adopted for investment management. Deep learning techniques (DL) has demonstrated notable advantages in addressing complex problems, such as the sparse IT. As was proposed by Zhang et al. [7], deep learning (DL) algorithms are capable of effectively extracting the intricate non-linear relationships between index components, thereby enabling the construction of a tracking portfolio using autoencoders (AE) as follows.

2.3. Autoencoders (AE).

Hinton and Salakhutdinov [17] stated AE as a feed-forward neural network employed for the purpose of dimensionality reduction and latent feature extraction. The AE compresses the input data into a latent representation using compact layers of hidden neurons, and then it reconstructs the original data using a decoder. Therefore, an AE consists of an encoder and a decoder process. According to Fig. 1, x represents the input data, $f(x)$ represents the encoder, which is an activation function forming a latent vector of hidden layers, z, and $g(z) = g(f(x))$ represents the decoder that generates the reconstruction x'.

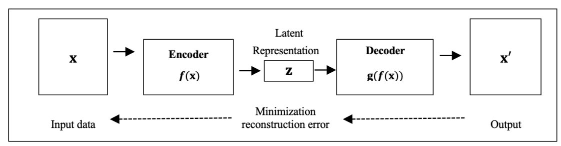


Fig. 1. General AE structure.

A distinctive feature of AEs is that they do not fully replicate the input data, but rather their latent properties. In this reconstruction, it is common for the dimension in z to be less than x. In the sense of investment portfolios construction, an EA can identify the minimum number of assets that explain the maximum information in all of them. This is the objective that is sought when the tracking portfolio is created. The AE learning process can be described as the minimization of the reconstruction error, denoted by $L(x, x')$, which is defined as the difference between x and x'. This is equivalent to the $\ell_2 - norm$.

$$\min \sum_{i=1}^{m} \left(x^{(i)}, x'^{(i)}\right) = \min \sum_{i=1}^{m} \left\|x^{(i)} - x'^{(i)}\right\|_2 \tag{5}$$

The minimization problem is solved using the backward propagation algorithm, which yields the model weights, w_1 and w_2. These weights are used to obtain the following expression.

$$\mathrm{z} = \mathrm{f}(w_1 \mathrm{x} + b_1) \text{ and } \mathrm{x}' = w_2 \mathrm{z} + b_2 \tag{6}$$

where, w_1 and w_2 represent the weights, while $f(\cdot)$ depicts the ReLU function and which one is given by $max(0, x)$. Zhang et al. [7] demonstrated that the ReLU function effectively addresses the gradient vanishing problem with a high speed of convergence compared to other activation functions. Furthermore, the researchers discovered that when the activation function is linear and the loss function is the mean squared error, the action of the single-hidden-layer under-complete AE is equivalent to principal component analysis (PCA). They demonstrated that EAs can generate a single value to describe each latent attribute. However, it is preferable to learn from a probability distribution for each latent attribute, rather than from a single value, to produce better generalization. Additionally, they discovered that this objective could be attained by implementing a generative model known as the variational autoencoder (VAE).

2.4. Variational Autoencoders (VAE).

VAEs represent a class of generative models that integrate elements of AEs and probabilistic models through the integration of Bayesian variational inference and deep learning [16]. In addition to learning the latent space representation, VAEs learn the probability distribution of the training data through a probabilistic mapping between data features and the latent space. In the VAE, the encoder generates the vector of parameters (θ), which one describes a distribution for each dimension of the latent space. It represents the variational posterior estimate, $p(z \mid x)$. In contrast, the decoder functions as a generative model represents the probability distribution $p_\theta(x|z)$. In this manner, a joint distribution can be defined as follows.

$$p(z|x) = p(x)p_\theta(x|z) \tag{7}$$

The learning process of a VAE can be described as the minimization of the reconstruction error of x, as is shown in Fig. 2.

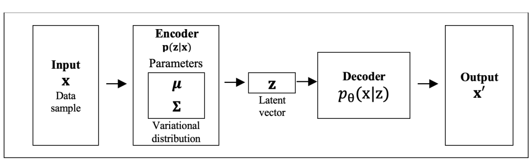


Fig. 2. VAE structure.

In the context of a normal distribution of the latent space, the encoder generates two vectors describing the mean, μ, and the variance, σ^2, while the decoder generates a latent vector z under a multivariate normal distribution with a diagonal covariance matrix. Following Zhang et al. [7] the variable z is set as input to the decoder, assuming a standard normal distribution, which one tries to reconstruct x as follows.

$$z = g(\varepsilon, \theta, x) = \mu + \sigma^2\varepsilon \tag{8}$$

where $\varepsilon \sim N(0, I)$. This setting ensures that the sampling results are derivable and that ε can propagate backward alongside the network. The loss function of the VAE is the negative log-likelihood, with a regularizer in the form of the Kullback-Leibler (KL) divergence. According to Kingma and Welling [16], the KL divergence is a measure of how one probability distribution is different from the other. The KL loss is equal to the sum of all the KL divergences between $x_i \sim N(\mu_i, \sigma_i^2)$, and the standard normal distribution. The evidence lower bound (ELBO) is a crucial concept in VAEs, which integrates the reconstruction loss and the KL divergence to balance the fidelity of data reconstruction with the regularization of the latent space. The ELBO is defined as follows.

$$ELBO = \mathbb{E}_{q(z|x)}[\log p_\theta(x|z)] - D_{KL}(p_i(z_i|x) \,||\, p(z_i)) \tag{9}$$

The loss function for the VAE, which one we seek to minimize, is the negative ELBO.

$$Loss = -ELBO = \mathcal{L}(\mathrm{x}, \mathrm{x}') + \lambda \textstyle\sum_i D_{KL}(p_i(z_i|x) \,||\, p(z_i)) \tag{10}$$

where $D_{KL} = \frac{1}{2}\left(-log\ \sigma_i^2 + \sigma_i^2 + \mu_i^2 - 1\right)$. The term $L(x, x')$ serves to penalize reconstruction errors. The additional term ensures that the learned latent-state distribution $p(z|x)$ is like the prior distribution $p(z)$, which minimizes the KL divergence between these two distributions. In addition, the hyperparameter λ regulates the relative weights of the function. As was stated by Zhang et al. [7], once the training is complete, the AEs generate an n-dimensional vector that includes n distinct latent factors. These factors are derived from the process of dimensionality reduction and can represent abstract and independent features of the input data. It is inevitable that compression or dimensionality reduction will result in information loss when the decoders reconstruct the input data from the latent representations.

The loss of information (L) represented in equation (11) shows the similarity between the input data and the data reconstructed after the encoding-decoding process, as was stated by Hinton et al. [17], so the goal will be for L to have the lowest value that reflects the least loss of information.

$$\mathcal{L} = \textstyle\sum_{i=1}^{m} \left\| x^{(i)} - x'^{(i)} \right\|_2 \tag{11}$$

3. Numerical application.

3.1. Data and methodology.

The sparse index-tracking problem is applied for the Nasdaq 100 Index (NDX) during January 2019 to December 2023 and the data was obtained from the Yahoo Finance API. The daily adjusted closing prices of 86 stocks of the index were used for the analysis. The remaining stocks were excluded from the analysis as they lacked complete information for the specified period.

Python and Keras were employed for data processing, implementation, and experimentation for the respective AE and VAE algorithms. Likewise, the designed experiments were implemented using a sliding window method, as is illustrated in Fig. 3. The sliding windows were employed to train the proposed models over a four-year period, with the testing period set to one month.

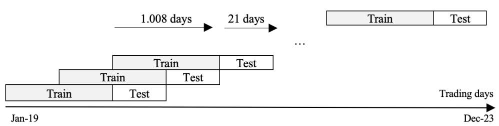


Fig. 3. Rolling training and testing windows.

In this sense, the rebalancing of portfolios is conducted monthly via the sliding windows. For each rebalancing period, the models were trained and tested. For example, the initial in-sample training period covered from January 2019 to December 2022, encompassing the first in-sample period of 1.008 observations, while the next 21 out-of-sample trading days. Then, the in-sample window is moved forward by 21 days and the new optimal tracking portfolio is determined using a window of 1.008 observations and again held unchanged for the next 21 out-of-sample days, and so on.

We settled the model based on the hyper-parameters of layers, the number of neurons, the ReLU activation function, optimizer, and regularization for optimal performance. To minimize the loss function, we use the Adam optimizer with learning rate given by 10^2. All the neural network policies are trained for 500 epochs. For every epoch, we test the learned policy on the validation set, and the best set of model parameters in the 50 tests is kept for out-of-sample testing.

3.2. Empirical results.

The empirical analysis started with a comparison of the proposed models based on the AE and VAE algorithms. Initially, a sliding window method was employed to compare the performance of both approaches, with a cardinality constraint of 43 assets. Fig. 4 illustrates the ETE for both algorithms over the 12-month period of 2023. It demonstrates the advantage of the VAE algorithm over the AE algorithm. The mean ETE for the VAE algorithm was 0.0075, while the AE algorithm achieved a mean ETE of 0.0083. In general, for the out-of-sample test months, the ETE for the VAE is lower.

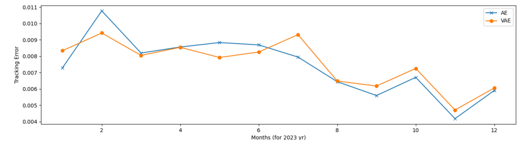
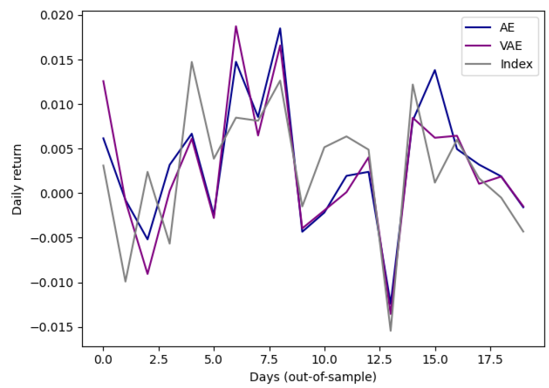
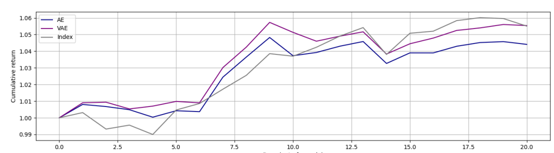


Fig. 4. Comparison of ETE results in the out-of-sample period.

Furthermore, Fig. 5 illustrates the first (month) out-of-sample performance of daily and cumulative returns for the two tracking portfolios and for the NDX index. It can be observed that the two methods, VAE and AE, exhibit close tracking. The VAE portfolio is closer than the AE portfolio. Once more, the relative advantages of the VAE-based tracking strategy are marked.



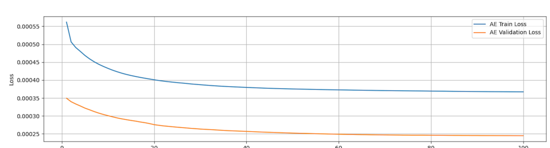
a. Daily return comparison – AE, VAE and benchmark



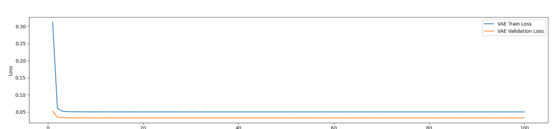
b. Cumulative return comparison – AE, VAE and benchmark

Fig. 5. Comparison of cumulative returns in the out-of-sample period.

However, this first experiment generated an overfitting problem. Fig. 6 shows the training and validation loss from AE and VAE.



a. AE training and validation loss.



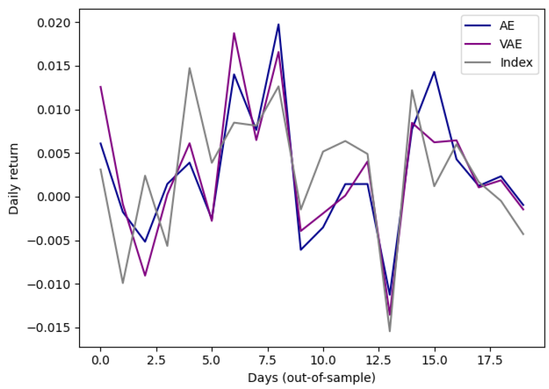
b. VAE training and validation loss.

Fig. 6. Training and validation loss: AE and VAE.

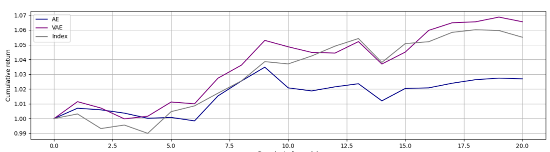
Given the overfitting problem of the first experiment, other two kind of experiments were proposed to solve it: dropout approach (experiment 2) and L2 regularization (experiment 3).

3.3. Dropout and L2 Regularization additional approaches.

The second experiment considered a dropout technique to prevent overfitting and where a dropout rate of 0.2 was settled. Fig. 7 shows the daily and cumulative return of AE, VAE and the benchmark on a comparison way.



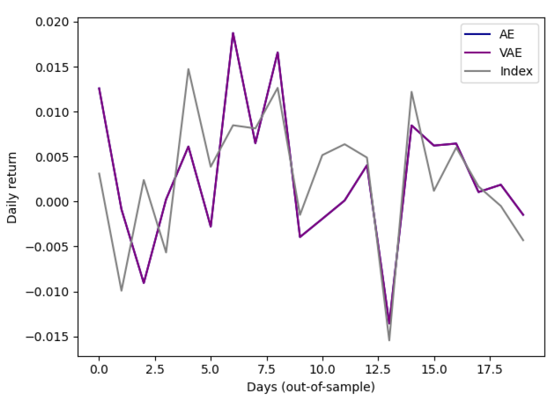
a. Daily return comparison – AE, VAE and benchmark



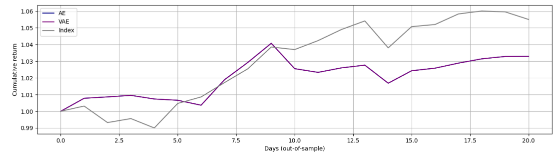
b. Cumulative return comparison – AE, VAE and benchmark

Fig. 7. Comparison of cumulative returns in the out-of-sample period (L2 regularization).

The third experiment for its part considered a L2 regularization technique to prevent overfitting and reaching a better generalization of the models. Fig. 8 shows the daily and cumulative return of AE, VAE and the benchmark on a comparison way.



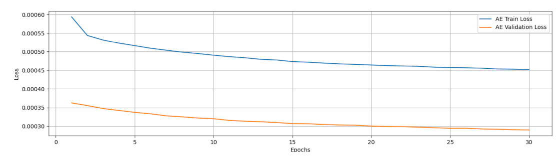
a. Daily return comparison – AE, VAE and benchmark



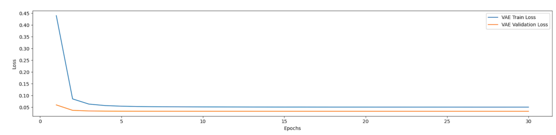
b. Cumulative return comparison – AE, VAE and benchmark

Fig. 8. Comparison of cumulative returns in the out-of-sample period (dropout regularization).

Regarding the metrics to evaluate the performance of a model during its training, Fig 9 depicts the training and validation loss of AE and VAE from the dropout regularization.



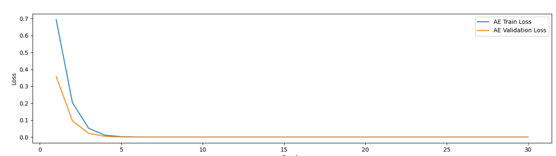
a. AE training and validation loss.



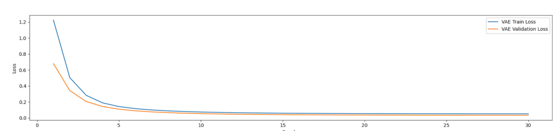
b. VAE training and validation loss.

Fig. 9. Training and validation loss: AE and VAE (dropout regularization).

Fig 10 for its part shows the training and validation loss of AE and VAE from the L2 regularization.



a. AE training and validation loss.



b. VAE training and validation loss.

Fig. 10. Training and validation loss: AE and VAE (L2 regularization).

It can be observed that only the additional experiment 3 (L2 regularization) leads to upgrade the performance of the models, at the same time the two models continue to present a good replication of the benchmark.

4. Conclusions.

The construction of tracking portfolios for sparse IT represents a challenging task due to the need of complex optimization algorithms. The practical constraints introduced, such as cardinality constraints, create a high-dimensional and computationally intensive problem. Two different deep learning methods such as AE and VAE were implemented to address this issue. The empirical analysis demonstrated that both AE and VAE-based models can effectively replicate the performance of the US Nasdaq 100 index while maintaining a limited number of assets in the portfolio. The results showed that both models replicated the benchmark index during in-sample and out-of-sample periods. However, the VAE model exhibited superior performance in terms of TE and cumulative returns, indicating its robustness and efficacy in handling the complexities of sparse IT. The AE model provided a suitable solution, but its performance was slightly lower regarding the VAE model, particularly under the out-of-sample validation. This suggests that the additional complexity and regularization inherent in the VAE model provide advantages in capturing the underlying data distribution and generalizing to new data.

The additional L2 regularization approach leads to upgrade the performance of both models, while the good tracking error performance was conserved.

Furthermore, the implementation of these models alleviates some of the computational issues associated with traditional optimization methods. By leveraging the strengths of deep learning, we were able to construct efficient and effective tracking portfolios that meet practical investment constraints. Future works could explore the integration of another DL algorithms, the impact of different market conditions such as of transaction costs, and the scalability of these models to another indices and asset classes.

References

1. Roll, R.: A mean/variance analysis of tracking error. The Journal of Portfolio Management, 18(4), 13-22 (1992). doi: 10.3905/jpm.1992.701922
2. Jorion, P.: Portfolio Optimization with Tracking-Error Constraints. Financial Analysts Journal, 59, 70-82 (2003). doi: 10.2469/faj.v59.n5.2565
3. Benidis, K., Feng, Y. and Palomar, D.: Sparse Portfolios for High-Dimensional Financial Index Tracking, IEEE Trans. on Signal Processing, 66(1), 155-170 (2018). doi: 10.1109/TSP.2017.2762286
4. Shu, L., Shi, F., Tian, G.: High-dimensional index tracking based on the adaptive elastic net. Quantitative Finance, 20(9), 1513-1530 (2020). doi: 10.1080/14697688.2020.1737328
5. Coleman, T. F., Li, Y., & Henniger, J.: Minimizing tracking error while restricting the number of assets. Journal of Risk, 8(4), 33 (2006).
6. Takeda, A., Niranjan, M., Gotoh, J. Y., & Kawahara, Y.: Simultaneous pursuit of out-of-sample performance and sparsity in index tracking portfolios. Computational Management Science, 10, 21-49 (2013). doi: 10.1007/s10287-012-0158-y
7. Zhang C, Liang S, Lyu F and Fang L.: Stock-Index Tracking Optimization Using Auto-Encoders. Front. Phys. 8:388 (2020). doi: 10.3389/fphy.2020.00388
8. Markowitz, H.: Portfolio selection. J. Fin., (7)1, 77-91 (1952).
9. Markowitz, H.: Portfolio selection: Efficient diversification of investments, New Haven, Yale University Press (1959)
10. Strub, O., Baumann, P.: Optimal Construction and Rebalancing of Index-tracking Portfolios. European Journal of Operational Research, 264(1), 370-387 (2018). doi: 10.1016/j.ejor.2017.06.055
11. Jansen, R., & Van Dijk, R. (2002). Optimal benchmark tracking with small portfolios. *Journal of Portfolio Management*, *28*(2), 33-39. doi:10.3905/jpm.2002.319830
12. Silva, J. C. S., & 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. doi: 10.1093/imaman/dpad007
13. Mutunge, P., & Haugland, D. (2018). Minimizing the tracking error of cardinality constrained portfolios. *Computers & Operations Research*, *90*, 33-41. doi: 10.1016/j.cor.2017.09.002
14. Feng, Y., & Palomar, D. P. (2016). A signal processing perspective on financial engineering. *Foundations and Trends® in Signal Processing*, *9*(1–2), 1-231.
15. Ouyang, H., Zhang, X., Yan, H.: Index tracking based on deep neural network. Cognitive Systems Research, 57, 107-114 (2019). doi: 10.1016/j.cogsys.2018.10.022
16. Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307-392.
17. Hinton, G. E. & Salakhutdinov, R.: Reducing the dimensionality of data with neural networks, Science 313(5786), 504–507, (2006). doi: 10.1126/science.1127647