Expert Systems With Applications 192 (2022) 116308

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An unsupervised learning framework for marketneutral portfolio

Salvatore Cuomo [a,*], Federico Gatta [a], Fabio Giampaolo [a], Carmela Iorio [b], Francesco Piccialli [a]

[a] Department of Mathematics and Application 'R. Caccioppoli', University of Naples Federico II, Italy
[b] Department of Economics and Statistics, University of Naples Federico II, Italy

ARTICLE INFO

Keywords:
Portfolio management
Arbitrage pricing theory
Time series
Cluster analysis

ABSTRACT

In this paper, we present a portfolio optimization strategy based on a novel approach in assets clustering on the financial background of the Arbitrage Pricing Theory, a well-known multi-factor model. In particular, our aim is to exploit data analysis tools, such as the techniques of features extraction and feature selection, to group assets that exhibit a significant exposition to the same risk factors. Then, we exploit the clustering to build a market-neutral portfolio and, more in general, an investment methodology that takes into account the peculiarities of the specific market. Finally, we apply our methodology in various case studies, discussing the results obtained and highlighting the strengths and the limits of the proposed strategy.

1. Introduction

Portfolio management is a process that starts from asset analysis to portfolio selection to outline the optimum portfolio possibilities that can be built from available investment opportunities. In this perspective, it is very important to be able to identify the underlying value factors that drive the returns of asset prices. Furthermore, with the growth of modern data analysis, this task has received a new direction of the investigation.

Let us consider a market made up of M assets $\Phi = \{(j, X^{(j)})\}_{j\in\{1,\dots,M\}}$ where j is the asset and $X^{(j)} = \{X_t^{(j)}\}_{t\in T}$ is the time series of its returns, with the time set common to all securities: $T = \{1,\dots,K\}$. Furthermore, we define a *portfolio* as a pair (Ω, Γ) with $\Omega \subset \Phi$ and Γ a vector of length equal to $|\Omega|$ such that the ith component represents the exposition on the ith asset of Ω. For simplicity, the total exposure is required to be unitary: $\|\Gamma\|_1 = 1$. Our proposal is twofold.

First, we partition Φ in such a way to group assets that exhibit a significant exposition on the same risk factors. Then, we create an appropriate portfolio that hedges exposure to those risk factors. Therefore, the selected portfolio can be defined as optimal according to a certain criterion. In other words, our aim is to find, between all the admissible hedged portfolios $\mathcal{P}$, the one that maximizes a certain function $\theta = \theta(\Omega, \Gamma)$:

$$P^* := (\Omega^*, \Gamma^*) = \underset{(\Omega,\Gamma)\in\mathcal{P}}{\operatorname{argmax}}\, \theta(\Omega, \Gamma) \tag{1.1}$$

To achieve this goal, we exploit as financial background the *Arbitrage Pricing Theory* (APT), since it takes into account a multiple factor model. Introduced by Ross (1976, 1977), the APT overcomes the drawbacks of the hypotheses underlying the Capital Asset Pricing Model (CAPM) proposed by Sharpe (1964) and independently developed by Lintner (1965) and Mossin (1966). The APT is described by an equilibrium formula based on the law of *one price* (i.e. two identical assets cannot have different prices). The methodological approach assumes that the returns of financial assets are related to a limited number of systematic factors (better known as *risk factors* and common to all financial securities) as well as to factors typical of individual companies or sectors, called idiosyncratic factors. The risk factors are indicated with F_i, $i = 1,\dots,n$ and are supposed to be random variables with zero mean and unit variance, uncorrelated with each other. According to this multi-factor asset pricing model, the explicit form of the return of asset j is:

$$X^{(j)} = \alpha^{(j)} + \beta_1^{(j)} F_1 + \cdots + \beta_n^{(j)} F_n + \varepsilon^{(j)} \tag{1.2}$$

The idiosyncratic risk is $\varepsilon^{(j)}$, while $\alpha^{(j)}$ represents its risk premium. The contribution of each risk factor F_i is weighted by $\beta_i^{(j)}$, which is a measure of the exposure of asset j on F_i. Different works exploit a linear model such as that in Eq. (1.2) to describe the asset returns. For example, see the works of Avellaneda and Lee (2010) and Pole (2011).

The core of our proposal is to group assets that exhibit a significant exposition on the same risk factors, and then create an appropriate portfolio that hedges the exposition on such risk factors. It should be noted that determining the correct risk factors in the APT model is a central task. Nowadays, with the growth of modern data analytics,

* Corresponding author.
E-mail addresses: salvatore.cuomo@unina.it (S. Cuomo), federico.gatta@unina.it (F. Gatta), fabio.giampaolo@unina.it (F. Giampaolo), carmela.iorio@unina.it (C. Iorio), francesco.piccialli@unina.it (F. Piccialli).

https://doi.org/10.1016/j.eswa.2021.116308
Received 26 April 2021; Received in revised form 17 November 2021; Accepted 25 November 2021
Available online 20 December 2021

this task has received a new direction of the investigation. In financial literature, this issue is a widely discussed open question.

In particular, the approaches proposed to identify the factors influencing the return of an individual asset fall into three main categories. The first one is the *macroeconomical approach*, that explains the return of a security through macroeconomic variables such as variations in Gross National Product or the rate of inflation (see for example the works of Chen, Roll, and Ross (1986) and Clare and Thomas (1994)). Another category is the *fundamental approach*, that is based on the fundamentals of the stock, meaning the approach leverages all the fundamental news about the company trading the securities (e.g. the BARRA model, explained by Sharpe, Alexander, and Bailey (1998)). Finally, the *statistical approach* searches for risk factors in the financial data themselves, applying the technique of feature extraction.

In what follows, we adopt the *statistical approach* in determining the correct risk factors in the APT model able to catch the collective performance in the data, rather than that of just a single stock. We want to highlight that we exploit the feature extraction to address this point and not to dimensionality reduction purposes. Furthermore, we observe that by following our approach the risk factors are constructed in such a way to eliminate redundancy in information, so their collinearity should be negligible. In particular, regarding the Principal Component Analysis (PCA) it is well known that the principal components are orthogonal to each others. Different researchers use PCA to address the collinearity problem (see for example Chatterjee and Hadi (2015)). Scholz, Fraunholz, and Selbig (2008) propose the hierarchical Neural Network Principal Component Analysis (h-NNPCA) to remove eventually correlation between input. In the Variational AutoEncoders (VAE) framework, the regularization factor in the layers and the request for minimizing the reconstruction errors should help in avoiding redundancy. To obtain this goal different works use the AutoEncoders (e.g. Bilal, Ullah, and Ullah (2019)). For this reason, in the proposed methodology we use both h-NNPCA and VAE as features extraction techniques.

The paper is organized as follows. Section 2 presents a literature review on the statistical approach to identify risk factor within a multi-factor model and clustering methods for financial purposes. Section 3 offers some background on the data analysis tools used in our proposal. Section 4 contains our proposal. Section 5 presents a discussion about some issues related to the choice of the strategy hyperparameters. In Section 6 we evaluate the performance of our methodology by analyzing real financial data belonging to three different markets: the Italian stocks market, Forex and Crypto. Section 7 concludes the paper with a discussion on further directions of analysis.

2. Literature review

Factors extraction

The literature shows an increasing interest in the statistical approach to identify the factors that influence the performance of an individual asset within the APT framework. For example, the proposal of Yip and Xu (2000) explains risk factors through Independent Component Analysis. Chiu and Xu (2004) in their paper proposed a new equilibrium model on the market based on the time–frequency analysis. Omran (2005) exploits the first three principal components as risk factors by explaining them from a financial point of view, that is linking them with the major sectors of the Egyptian economy. Spyridis, Sević, and Theriou (2012) presents a comparison between the macroeconomic and the statistical approaches, while Tzagkarakis, Caicedo-Llano, and Dionysopoulos (2013) introduces a nice pure alpha strategy based on the Probabilistic Principal Factors Analysis. It is worth noting the work done by Ladrón de Guevara and Torra in a group of three articles published between 2014 and 2019. In particular, de Guevara Cortés and Porras (2014) attempts to explain risk factors through PCA or Factor Analysis, exploiting a two-steps strategy in order to determine the coefficients of the model and risk-premium for each factor, while Ladrón de Guevara Cortés, Torra Porras, and Monte Moreno (2018) is related to the extraction of risk factors through the Independent Component Analysis. Ladrón de Guevara Cortés, Torra Porras, and Monte Moreno (2019) explains risk factors with the Neural Network Principal Component Analysis (NNPCA). These authors show the effectiveness of the NNPCA in the risk factors extraction. This is due to the capability of NNPCA to preserve some useful properties of the standard PCA, as we recall in Section 3.

Regardless of the risk factors extraction approach, it is pointed out by several studies that the idiosyncratic risks contains useful information. For example, Blitz, Huij, and Martens (2011) use the residuals from the factor model proposed by Fama and French (1993) to generate a momentum strategy, showing that there are some patterns contained in the $\varepsilon^{(j)}$. Instead, Imajo, Minami, Ito, and Nakagawa (2020, 2021) exploit the residuals from the factor model with a statistical approach together with neural networks showing the effectiveness and the robustness of their portfolio optimization proposal over the last two decades, characterized by three financial crises.

Starting from a factor model, a wide literature has been developed regarding the possibility of deleting the exposure on the risk factors. A strategy that tries to eliminate such exposition belong to the class of statistical arbitrage strategies and in particular to the *market-neutral arbitrage* strategies. Several works claimed the profitability of this type of trading strategy. For example, Visagie and Hoffman (2017) examin the performance of statistical arbitrage systems over a range of different financial markets, whereas Avellaneda and Lee (2010) exploit a linear factor model and then compare the profits gained using as risk factors those obtained via PCA (statistical approach) and those obtained exploiting the Exchange-Traded Fund (macroeconomical approach). The authors show the profitability of the strategy when using the statistical approach. Another detailed description about the market-neutral arbitrage, which shows the pro and cons of this type of trading strategy, can be found in Nicholas (2000). Finally, we propose to use a market-neutral strategy since our aim is not to obtain high expected returns, but to reduce the risks, bearing in mind that the market operators are risk-adverse and that financial crisis, such as that of March 2020, is right around the corner.

Time series clustering

Due to the recent interest in data evolving over time, time series clustering has become an important topic. Regarding the literature about time series clustering, a rich survey can be found in Aghabozorgi, Shirkhorshidi, and Wah (2015) and Liao (2005). These surveys after showing the difficulties of time-series clustering against that on the so-called *static data* (especially related to the temporal evolution of the series that could change their peculiarities and, so, walk up or walk away from each other), expose the state of the art of the clustering techniques proposed in the literature. In particular, Liao divides the existing techniques into three main groups: *raw-data-based*, *feature-based* and *model-based*. The first group works directly with the time series (see, e.g. De Luca and Zuccolotto (2011) and Košmelj and Batagelj (1990). The feature-based algorithms extract some features from the series and then cluster these features (see, e.g. Alonso and Maharaj (2006), Maharaj, D'urso, and Galagedera (2010) and Van Wijk and Van Selow (1999)). The model-based algorithms classify the estimated coefficients of a model fitting the original time series (see e.g. Corduas and Piccolo (2008), Iorio, Frasso, D'Ambrosio, and Siciliano (2016), Otranto (2008) and Piccolo (1990)). In the last years, clustering techniques dealing with financial time series aimed to portfolio management as well as portfolio selection have been proposed. To cluster financial time series, a procedure based on a feature-based algorithm was proposed by Fu, Chung, Ng, and Luk (2001). In the survey proposed by Aghabozorgi and Teh (2014), more emphasis is placed on the multi-step algorithms.

For example, the work of Lai, Chung, and Tseng (2010) exploits a two-steps procedure based on two different time granularities, while the proposal of Aghabozorgi and Teh (2014) is based on a three-step procedure applied to the stock market. Nair, Kumar, Sakthivel, and Vipin (2017) introduce a nice application of stocks clustering in the development of a trading strategy. In order to suggest a strategy to build a portfolio of stocks, Iorio, Frasso, D'Ambrosio, and Siciliano (2018) propose to use a parsimonious time series clustering approach within the framework of financial time series. Finally, another application of clustering in finance, can be found in Nakagawa, Kawahara, and Ito (2020), where the authors exploit a modified version of the famous k-Means algorithm, namely the *x-Means++*, to group assets and then balance the risk contribution between clusters.

For a detailed literature review about clustering methods for portfolio selection one can refer to Iorio et al. (2018) and the references therein.

Our clustering strategy is a feature-based one, in which the features correspond to the risk factors that affect the returns of each asset (as we see later, this procedure implies some pros and cons), and the portfolio is built canceling such exposition.

3. Background on features extraction and selection

3.1. Features extraction

In the construction of our proposal, the first stage is the features extraction.

Let us consider a group of M time series with K observations each one, that is $X^{(j)} = \{X_t^{(j)}\}_{t\in\{1,\dots,K\}}$ $\forall j \in \{1,\dots,M\}$. Then, we use the features extraction to obtain n time series that are able to explain the general trend of the data and not only that of the individual series. In literature, there exist various techniques used to achieve our goal. We exploit three of them. The first one is the widely used PCA. The others are h-NNPCA and VAE. Since the PCA is a well-known feature extraction algorithm, in the next subsections, we pay attention to expose the others.

Hierarchical neural network principal component analysis

The Neural Network Principal Component Analysis is a non-linear version of the widely used PCA founded on the Neural Networks (NN). There exist several versions of NNPCA: standard, circular and inverse, but for our purpose, the best one is the h-NNPCA, proposed by Scholz and Vigário (2002) and largely discussed by Scholz et al. (2008).

In the following we apply the h-NNPCA, since it is best suited for our problem than the other NNPCA types because it retains two desirable properties of the PCA. The first one is *scalability*, i.e. the first $p \leq n$ PCs form the p-dimensional subspace that catches the maximal variance of the data. The second one is *stability*, i.e. if we consider another $\tilde{n}$ decomposition, then the pth components of each decomposition are the same, for each $p \leq min(n, \tilde{n})$.

In order to obtain the h-NNPCA, we use a dense NN, that can be considered as a collection of:

- $l+1$ vectors L_j, $j \in \{0,1,\dots,l\}$ called *layers*, each one made up of n_j elements, also known as *neurons*, that is $L_j \in \mathbb{R}^{n_j}$.
- l weights matrices and activation functions $W_j \in \mathbb{R}^{n_j \times n_{j-1}}$, ρ_j with $j \in \{1,\dots,l\}$ such that $\forall j$ it results:

  $$L_j = \rho_j(W_j L_{j-1})$$

- A *loss function*, that measures the distance between the layer L_l and a target vector, which is used to compute the weights matrices, i.e.:

  $$(W_1,\dots,W_4) = \underset{\mathcal{W}}{\operatorname{argmin}} \frac{1}{K}\sum_{t=1}^{K} E(X_t; W_1,\dots,W_4)$$

  with $\mathcal{W} = \mathbb{R}^{n_1\times n_0} \times \cdots \times \mathbb{R}^{n_4 \times n_3}$.

In the h-NNPCA, both the input layer and the output one have the same size, equal to the number of series. We use five layers. The central one has dimension n equal to the number of time series that we want to extract, also known as *Principal Components* (PCs). This architecture is known as the *bottleneck network*. Fig. 1 clarifies the architecture.

Our purpose is to predict, time by time, the vector of asset returns $X_t = [X_t^{(1)},\dots,X_t^{(M)}] \in \mathbb{R}^M$ exploiting the same vector as input. Regarding the loss function, to obtain a kind of order between the principal components, we have to take into account, for each $k \in \{1,\dots,n\}$, the Mean Square Error (MSE) calculated on the sub-network in which the central layer is made up only by the first k neurons: $E_k(X_t; W_1^{(k)},\dots,W_4^{(k)})$ where $W_1^{(k)},\dots,W_4^{(k)}$ are the corresponding sub-matrices of $W_1,\dots,W_4$. Finally, the total error is the sum of the single ones:

$$E(X_t; W_1,\dots,W_4) = \sum_{k=1}^{n} E_k(X_t; W_1^{(k)},\dots,W_4^{(k)})$$

The h-NNPCA is performed by using the MATLAB® toolbox provided by Scholz (2006).

Variational AutoEncoders

Another features extraction technique that we exploit is the VAE, described in detail by Kingma and Welling (2019). A VAE is a neural network whose inputs and outputs overlap, obviously with at least a hidden layer of dimension strictly less than the input dimension (otherwise the problem is trivial and, clearly, without any practical usefulness). It is made up of two pieces: an *encoder* who projects the data in a lower-dimensional space, also known as *latent space*, and a *decoder* that tries to reconstruct original data from the representation provided in latent space, working with a loss function such as MSE. In particular, in order to obtain a sort of regularization, the encoder gives us not a deterministic point of the latent space, but rather a distribution over it (a normal one, so we have to specify two values: mean μ and standard deviation σ). The loss function takes into account not only the distance between the real data and the predicted ones but also a regularization term representing the distance between the extracted latent distribution and that of the latent variable given the reconstructed data (measured in terms of the Kullback–Leibler divergence). Finally, we point out that although the architecture of these networks is absolutely general because we work with time series, for our purpose is convenient to use a technique that is able to catch the temporal dependence between observations, in particular the Long Short-Term Memory. In Fig. 2 are displayed the operations of VAE.

Empirical applications of the VAE in finance can be found in Montesdeoca, Squires, and Niranjan (2019) and in Mancisidor, Kampffmeyer, Aas, and Jenssen (2021).

3.2. Features selection

The second step of our proposal is the selection of the extracted features. In particular, we need a technique that, starting from a linear model, is able to identify the subset of exogenous that significantly affect the endogenous. Although the Least Absolute Shrinkage and Selection Operator (LASSO) proposed by Tibshirani (1996) is an estimator widely used for features selection, sometimes it could result inappropriate. In fact, for a feature selection technique, it is desirable that it satisfies the so-called *oracle properties*, formalized by Fan and Li (2001). The starting point is a simple model with K observations and n inputs: $X_t^{(j)} = \sum_{i=1}^{n} F_{i,t}\beta_i^{(j)} + \varepsilon_t^{(j)} \quad \forall t = 1,\dots,K$, where $F = [F_i]_{i\in\{1,\dots,n\}}$ is the inputs matrix, $\beta^{(j)}$ is the coefficients vector, $X^{(j)}$ is the output vector and $\varepsilon^{(j)}$ is an additive contribution representing the error term. We assume that:

- The errors are i.i.d. normal random variables with 0 mean and σ^2 variance.
- Only $F_1,\dots,F_{\tilde{n}}$ are relevant, i.e. $\mathcal{A} = \{1,\dots,\tilde{n}\}$ and $\beta_i^{(j)} = 0 \quad \forall i \notin \mathcal{A}$.

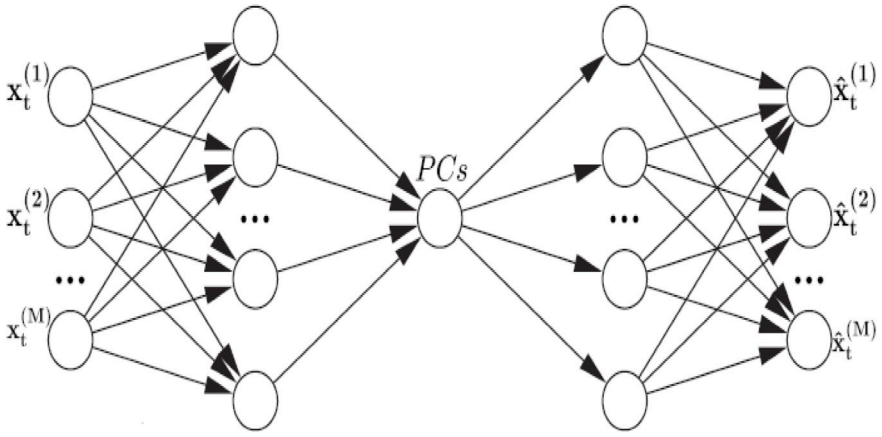

Fig. 1. Bottleneck Architecture of the h-NNPCA. The first and the last layers embodies the time series data time by time, while the central layer provides the Principal Components extracted.

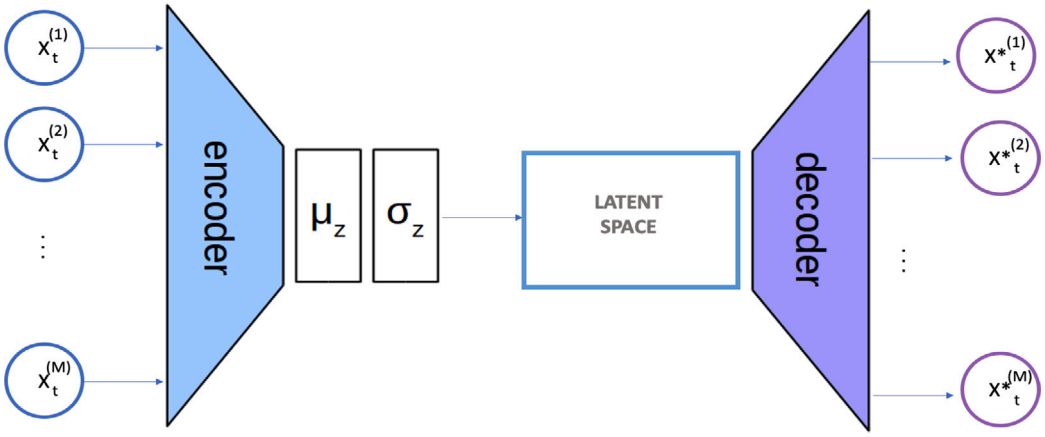

Fig. 2. VAE Architecture. Starting from the data, we obtain the mean and variance of the normal distribution on the latent space, then we use a randomly sampled point to reconstruct the original data.

- Define $\frac{1}{K}F^TF = C = \begin{bmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix}$ *with* $C_{1,1} \in \mathbb{R}^{\bar{n}\times\bar{n}}$.

An estimator $\hat{\beta}_K^{(j)}$ that treats as relevant the variables in $\mathcal{A}_K$, satisfies the *oracle properties* if:

- *Consistency in Variable Selection*, i.e. $\lim_{K\to\infty}\mathbb{P}(\mathcal{A}_K = \mathcal{A}) = 1$, that is the estimator must asymptotically identify the correct subset of relevant exogenous.
- *Asymptotic Normality*, i.e. $\sqrt{K}(\hat{\beta}_K^{(j)} - \beta^{(j)}) \to_d N(0, \sigma^2 C_{1,1}^{-1})$, that is the regression provided must asymptotically be as good as that done knowing in advance the correct subset of relevant inputs.

The LASSO drawback is the inconsistency. In fact, various works such as that of Meinshausen, Bühlmann, et al. (2006), show that often the regularization parameter chosen to attempt to provide the best prediction tends to treats as relevant, variables that are just noise. However, it was proposed by Zou (2006) a modified version of LASSO, namely the *Adaptive Least Absolute Shrinkage and Selection Operator* (A-LASSO), which satisfies the *oracle properties*. The A-LASSO estimator is the one that minimizes the following loss function:

$$\underset{\beta^{(j)}}{\text{argmin}} \left[\frac{1}{K}\|X^{(j)} - F\beta^{(j)}\|_2^2 + \lambda\|W\beta^{(j)}\|_1 \right] \tag{3.1}$$

where λ is the regularization parameter and $W \in \mathbb{R}^{n\times n}$ is a diagonal matrix of weights $w_1, \dots, w_n$. Regarding their choice, Zou (2006) proves that choosing $\{\lambda_K\}_{K\in\mathbb{N}}$ such that $\lim_{K\to\infty}\lambda_K K^{-1/2} = 0$ and $\lim_{N\to\infty}\lambda_K K^{(\gamma-1)/2} = \infty$ and $w_i = |\hat{\beta}_{OLS,i}|^{-\tau}$, where $\hat{\beta}_{OLS}$ is the Ordinary Least Squares (OLS) unbiased estimator and $\tau > 0$, then the A-LASSO estimator satisfies the *oracle properties*. So with A-LASSO, we have two parameters to optimize: λ and τ.

In this work, we use A-LASSO to feature selection, since it has different pros. It is easily implementable and the computational times are quite fast. As pointed out by Zou (2006), the A-LASSO can be obtained as a LASSO estimator with input matrix: $\tilde{F} = FW^{-1}$ and the computational time is the same order of the OLS. Moreover, this technique of feature selection is widely used in financial applications. Recently, Panagiotidis, Stengos, and Vravosinos (2018) exploit A-LASSO in order to understand the factors that affect the Bitcoin returns.

Although these pros, A-LASSO has a big drawback, that is the presence of collinearity between regressors could harm its performance, as pointed out by Zou and Zhang (2009). However, in the proposed methodology, the collinearity is not an issue. Indeed, we already stated in Section 1 that, at least from a theoretical point of view, there should not be any collinearity between the risk factors used as explanatory variables in the A-LASSO. Nonetheless, we test this assumption, exploiting the condition number test for collinearity, as reported by Kim (2019). The test consists estimating of the conditioning number of the square matrix obtained considering the matrix product between F^T and F, where F^T is the transpose of F that is the matrix in $\mathbb{R}^{N\times n}$ having

Table 1
K values for the Kim test in the Forex and FTSE MIB datasets. Values lower than 10 indicates the absence or the presence of negligible collinearity.

	PCA 5 PCs	PCA 6 PCs	NNPCA 5 PCs	NNPCA 6 PCs	VAE 5 PCs	VAE 6 PCs
Forex	1.0	1.0	1.0	1.1	2.0	2.2
Italian Market	1.0	1.0	1.0	1.0	3.0	3.3

Table 2
K values for the Kim test in the Crypto dataset. Values lower than 10 indicates the absence or the presence of negligible collinearity.

	PCA 4 PCs	PCA 5 PCs	PCA 6 PCs	NNPCA 4 PCs	NNPCA 5 PCs	NNPCA 6 PCs	VAE 4 PCs	VAE 5 PCs	VAE 6 PCs
Crypto	1.0	1.0	1.0	1.2	1.0	1.1	4.0	4.1	3.7

as columns the standardized risk factors considered. Once obtained the maximum and minimum eigenvalues of $F^T F$, respectively $\lambda_{max}, \lambda_{min}$, the conditioning number is defined as $K = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$. If $10 \leq K \leq 30$, there is collinearity, while if $K > 30$, there is strong collinearity. The results shown in Table 1 and in Table 2 confirm the presence of null or negligible collinearity in all the datasets employed in the following experimental Section 6. To conclude, we overcome the collinearity drawback of A-LASSO estimator.

Finally, we want to reiterate that the estimator just discussed is used principally for features selection and not for regression. The parameters can be estimated separately for each temporal window. In order to estimate the coefficients separately for each assets we suggest to divide the data into several temporal windows (according to the temporal horizon of investment). For each one, we performed a pooled OLS regression. Finally, for each asset the coefficients of the model are simply computed as the average of the coefficients in each temporal window. Formally, we divide our observations into H temporal windows and for each one we perform a pooled OLS in order to obtain the coefficients $\hat{\beta}_h^{(j)}$. Thus, the final estimate of model coefficients can be expressed in the following way:

$$\hat{\beta}^{(j)} = \frac{1}{H} \sum_{h=1}^{H} \hat{\beta}_h^{(j)} \omega_h$$

where ω_h is the proportion of stocks falling in the h-th temporal window.

4. A framework for the portfolio optimization

The proposed portfolio optimization strategy can be summarized as follows:

1. perform a feature extraction algorithm on financial time series to obtain the risk factors;
2. from the APT model compute a A-LASSO estimator to select the features;
3. perform a cluster analysis on the selected features;
4. build a market neutral portfolio.

Fig. 3 shows a summary scheme.

To reach our goal, we define a set of possible hyperparameters $\Theta = \prod_{j=1}^{7} \Theta_j$. Each hyperparameter is a vector $\theta = (\theta_1, \ldots, \theta_7)$, whose components are defined in this way:

- θ_1 is the statistical technique used to extract risk factors from the data.
- θ_2 is the number of risk factors considered.
- θ_3 indicates the scaler used in the pre-processing stage.
- θ_4 represents the time horizon of the investment, expressed in months.
- θ_5 is the length of the possible portfolios extracted from each class (it can be a function of the class itself).
- θ_6 represents the criterion that our portfolios try to optimize. For example, we can exploit the sum of α of the resulting portfolio, or we can use Sharpe Ratio.
- θ_7 is a number in $\{-1, 1\}$ that is multiplied by the weights vector. In particular 1 means that we invest in agreement with the selected portfolios, while -1 means that we are investing against the selected portfolios. In other words, in the first case, we are assuming a kind of momentum, while in the second case we are considering a mean-reverting property.

4.1. Assets clustering

Our aim is to construct a feature-based clustering algorithm. To do this we extract from the time series a vector of features, that is a function $A : \Phi \rightarrow \{0, 1\}^n$ where $A(j) = I_{\mathcal{A}_j}(PCs)$ and $I_{\mathcal{A}_j}$ is the indicator function of the set $\mathcal{A}_j$ of the PCs that, according to the features selection, affect the return of asset j. Furthermore, following the classifications reported in the above-mentioned surveys, ours is a *partitioning crisp method* (that is we construct a partition of Φ and then each cluster is identified with an element of the partition) that acts on the *structure level* (i.e. we study the global structure and not local patterns). Algorithm 1 summarizes the clustering procedure (observe that, for simplicity, we consider a cluster as a pair $(\mathcal{A}, \Omega)$ made up of the set of principal components that affect the returns of the securities in the cluster and the subset of Φ of the assets in the cluster). In the following, we comment on this algorithm in detail.

Let us consider market Φ, **line 2**. We have the time series of the returns of M assets with K observations each one and we assume that there are not missing values. Now, in **line 3** we apply the feature extraction technique (let us indicate it with θ_1) in order to extract $n = \theta_2$ risk factors to be used in the APT, then in **lines 4–6** we apply the θ_3 scaler. The peculiarity of this work is that we use a big θ_2 and then we perform feature selection. The underlying idea is that while some risk factors are common to almost all assets (for example the market), other risk factors could definitely affect the performance of some assets but not those of other ones.

Afterwards, for each asset, we apply a procedure of features selection performed through A-LASSO. We perform A-LASSO working on inputs exogenous constituted only by the principal components (for the clustering, we do not take into account the constant).

The initial weights are $|\beta_{OLS}|^{-\tau}$, where $|\beta_{OLS}|$ is the OLS estimate. The two coefficients of the model (τ for the weights and λ for the strength of the regularization) are chosen in **lines 9–11** through grid-search applied to 3-fold nested cross-validation on MSE, considering as valid only those couples such that the number of principal components saved from the selection is between 2 and 4 (**lines 12–14**), in order to avoid both models too complex or regularizations too aggressive. The *S-fold nested cross validation* is a widely used technique that provides an estimate of the prediction error (in our case of the MSE). It consists of generating S subsets from the train set $T = \{1, \ldots, K\}$ according to the temporal dimension, that is $T_s = \{1, \ldots, K_s\}$ with $1 < K_1 < \cdots < K_S = K$, and then split them into train and test subsets: $Tr_s = \{1, \ldots, TR_s\}$ and $Te_s = \{TR_s + 1, \ldots, K_s\}$. The estimate is the mean of the error

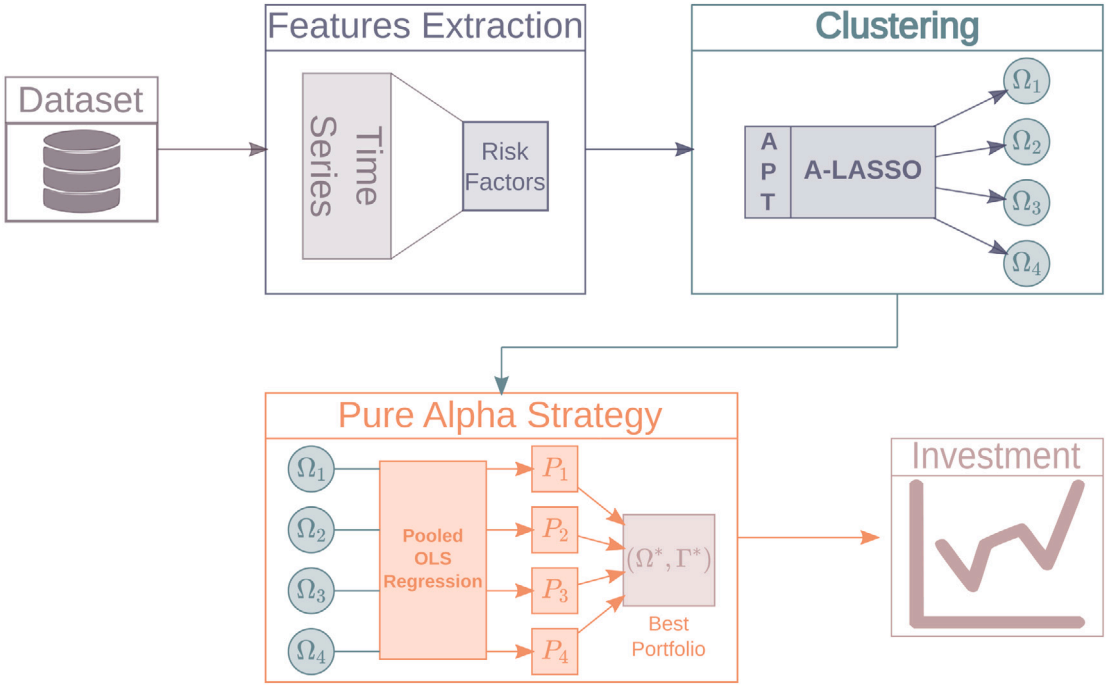

Fig. 3. Summary Scheme — We exploit a features extraction technique to extract the risk factors, then through the APT and A-LASSO we build the clusters. Within each cluster, we construct a market-neutral portfolio, then the most promising one is selected for the investment.

committed in the S test subsets Te_s, after training the model on the corresponding train subset Tr_s.

Algorithm 1 Clustering Algorithm

1: {*Define the dataset, extract PCs and scale them.*}
2: Define $\mathbf{\Phi} = \{1, ..., M\}$ and $\mathbf{X} = [X^{(1)}, ..., X^{(M)}]$
3: $PCs = [PC1, PC2, ..., PC\theta_2] = \theta_1(\mathbf{X}, \theta_2)$
4: **for** $PC \in PCs$ **do**
5: $PC = \theta_3(PC)$
6: **end for**
7: {*Find the best* λ *and* τ *through Cross-Validation and Grid-Search.*}
8: Define Λ and $\mathcal{T}$
9: **for** $j \in \mathbf{\Phi}$ **do**
10: **for** $(\lambda, \tau) \in \Lambda \times \mathcal{T}$ **do**
11: Compute $\mathcal{A}_j^{\lambda,\tau} =$ **A-LASSO**$(X^{(j)}, PCs; \lambda, \tau)$
12: **if** $2 \leq |\mathcal{A}_j^{\lambda,\tau}| \leq 4$ **then**
13: Compute $mse^{\lambda,\tau} =$ **Cross-Validation**$(X^{(j)}, PCs; \lambda, \tau)$
14: **end if**
15: **end for**
16: Save $(\lambda, \tau) = \text{argmin}_{(\lambda,\tau)\in\Lambda\times\mathcal{T}}\, mse^{\lambda,\tau}$ and the corresponding $\mathcal{A}_j$.
17: **end for**
18: {*Create Clusters.*}
19: Initialize $\mathcal{C} = \emptyset$
20: **for** $j \in \mathbf{\Phi}$ **do**
21: **if** $(\mathcal{A}_j, \Omega) \in \mathcal{C}$ **then**
22: Update $\Omega = \Omega \cup \{j\}$
23: **else**
24: Update $\mathcal{C} = \mathcal{C} \cup (\mathcal{A}_j, \{j\})$
25: **end if**
26: **end for**

Once we complete our research, what we obtain for each asset is a linear model such as that in Eq. (1.2), with some $\beta_i^{(j)}$ coefficients equal to 0. After defining in **line 16**, asset by asset, the set of the principal components whose coefficients are different from zero $\mathcal{A}_j = \{i \in \{1, \dots, n\}\ s.t.\ \beta_i^{(j)} \neq 0\}$, we introduce an equivalence relationship between assets in this way: $(j, X^{(j)}) \sim (l, X^{(l)}) \iff \mathcal{A}_j = \mathcal{A}_l$. In other words, two assets are equivalent if and only if, they share the same risk factors. Then, the clusters are identified, in **lines 19–26**, with the equivalence classes of $\sim$ relationship.

We point out that two of the strengths of our proposal are that we do not need to set the number of clusters in advance and we neither need to define a similarity function in the construction stage, which, according to Aghabozorgi et al. (2015), are quite difficult tasks when working with real problems. However, the backlash is that we have to face up with the determination of the A-LASSO hyperparameters. Furthermore, we consider only the first principal components, and this can help us in cleaning the data from noise when clustering, even if setting a proper number of principal components to consider could be difficult. However, in our opinion, the most important advantage of our clustering methodology is that it is immediately usable in the construction of an investment strategy, even if not all the obtained clusters are relevant from this point of view.

Finally, in order to provide a real example, let us consider the clusters obtained in the Italian stock market, extracting 6 risk factors through h-NNPCA, with a standard scaler. We obtain 16 classes, and the first 8 are reported in Table 3. Observe that, out of the 16 clusters, only 3 can be practically used in the construction of an investment methodology (classes 4, 7 and 8), as we will explain in the next subsection. Furthermore, observe that, as a consequence of the scalability property, the first principal component is the most important one and it affects the returns of almost all the stocks considered, while the other components have a minor impact, that sometimes is negligible.

4.2. Market-neutral portfolio

In order to obtain the investment strategy, firstly we explicit our assumptions whose are of various nature: there is a theoretical assumption, i.e. we suppose that for each asset in the cluster is valid a representation of type (1.2) with the principal components as risk factors; there are practical assumptions i.e. we assume that there are no transaction costs, all the assets are tradable both long than short, and all the considered assets are infinitely divisible.

Now, let us consider the APT equation for a generic asset j, taking into account the features selection stage:

$$X^{(j)} = \alpha^{(j)} + \sum_{i \in \mathcal{A}_j} \beta_i^{(j)} F_i + \varepsilon^{(j)}$$

For the estimation of the parameters, we use the Pooled OLS Regression previously described. We set the length of temporal windows according to the temporal horizon of the investment, which we indicate with θ_4.

Then, let us fix class $(\tilde{\mathcal{A}}, \tilde{\Omega})$, assume that the assets in this class are characterized by $\tilde{n}$ risk factors and consider a portfolio made up of $\theta_5 > \tilde{n}$ assets in this class, that is (Ω, Γ) where, without loss of generality, we assume:

$$\Omega = \{(1, X^{(1)}), \dots, (\theta_5, X^{(\theta_5)})\} \subset \tilde{\Omega} \quad \textit{and} \quad \Gamma = (\gamma^{(1)}, \dots, \gamma^{(\theta_5)})$$

The generic portfolio constructed in this way is represented in Eq. (4.1).

$$X^{(\Omega)} = \sum_{j=1}^{\theta_5} \gamma^{(j)} X^{(j)} = \sum_{j=1}^{\theta_5} \gamma^{(j)} \alpha^{(j)} + \sum_{i \in \tilde{\mathcal{A}}} \left(\sum_{j=1}^{\theta_5} \gamma^{(j)} \beta_i^{(j)} \right) F_i + \sum_{j=1}^{\theta_5} \gamma^{(j)} \varepsilon^{(j)} \tag{4.1}$$

If we adequately choose portfolio weights, then we obtain a market-neutral portfolio, described in Eq. (4.2).

$$X^{(\Omega)} = \sum_{j=1}^{\theta_5} \gamma^{(j)} \alpha^{(j)} + \sum_{j=1}^{\theta_5} \gamma^{(j)} \varepsilon^{(j)} = \alpha^{(\Omega)} + \varepsilon^{(\Omega)} \tag{4.2}$$

The weights of an admissible portfolio of this type satisfy Eqs. (4.3):

$$\sum_{j=1}^{\theta_5} \gamma^{(j)} \beta_i^{(j)} = 0 \quad \forall i \in \tilde{\mathcal{A}} \tag{4.3}$$

These equations form a homogeneous linear system of $\tilde{n}$ linear equations in θ_5 variables $\gamma^{(1)}, \dots, \gamma^{(\theta_5)}$, so there exist infinite solutions. Thus, we should introduce a criterion θ_6 and we should try to find the weights vector that optimizes the estimate of this criterion provided by the 3-fold nested cross-validation). Then, we invest according to the thus obtained portfolio or against it, in other words, if we indicate such portfolio with p, then we invest in $\theta_7 \cdot p$ with $\theta_7 \in \{-1, 1\}$. In other words, we are trying to catch the peculiarities of $\alpha^{(\Omega)}$ and $\varepsilon^{(\Omega)}$ in order to select, between the different market-neutral portfolios, those that could be profitable in the future. In fact, as we already pointed out in Section 2, the APT residuals could hide some kind of pattern useful for the investment strategy.

The process is summarized in algorithm 2.

Finally, it is worth noting that we select the three best portfolios and not only one. In this way, we are improving diversification as well as trying to eliminate possible errors in our portfolios.

Algorithm 2 Investment Methodology

```
{Create the universe of possible portfolios}
Initialize 𝒫 = ∅.
for (Ã, Ω̃) ∈ C do
    Define 𝒫_temp = C(Ω̃, θ5) as the set of combinations without repetition of θ5 elements in Ω̃.
    Update 𝒫 = 𝒫 ∪ 𝒫_temp
end for
{Find the portfolios to invest in}
for P ∈ 𝒫 do
    for j ∈ P do
        Compute α(j), β(j) = Pooled-OLS(X(j), 𝒜_j; θ4)
    end for
    Solve the optimization problem to obtain weights vector Γ.
    Compute θ6^P = Cross-Validation(P, Γ)
end for
Find the 3 argmax_{P∈𝒫} θ6^P and save them: P̃1, P̃2, P̃3
Invest in θ7 · P̃j   ∀j ∈ {1, 2, 3}
```

Table 3
Clusters results in Italian Market — hNNPCA 6 risk factors standard scaler.

Eq Class	PC	n_stocks	Eq Class	PC	n_stocks
1	[1, 2, 3, 4]	2	5	[1, 2, 5]	2
2	[1, 4, 6]	1	6	[1, 4, 5, 6]	2
3	[1, 3, 4]	2	7	[1, 6]	5
4	[1, 4]	5	8	[1, 5]	3

5. Implementation details

In this section, we briefly clarify some issues related to the practical implementation of our methodology. In the first subsection, we discuss some details connected to the functioning of the hyperparameters, while in the second one we expose our methodology for their determination.

5.1. A learning approach

Regarding θ_3, note that, in order to satisfy the hypothesis of the APT, that is our financial background, we should standardize the principal components. Furthermore, what we obtain is a so-called *nonlinear whitening transformation* that, how reported in Scholz et al. (2008), removes the nonlinearities in the data and allows us to handle the principal components thus obtained with a linear method. However, from a practical point of view, as long as we delete the exposition on the risk factors, we can also exploit other scalers.

As for θ_5, the strategy to obtain the weights for the hedged portfolio while we work with $|\mathcal{A}| + 1$ assets is quite straightforward because the number of hedged portfolios is finite. Instead, when we work with a bigger number of assets, say m, we have to exploit some optimization technique. In this experiment, we use *Sequential Least Squares Programming* (*SLSQP*), and we impose both linear and non-linear constraints. The latter regard the condition that the l_1 norm of weights vector is equal to 1 (furthermore this condition also gives us the domain in which to search the optimal weights: the hypercube $[-1, 1]^m$); the former concerns the exposition to risk factors, that is requested to be 0 for each one. As for the initial weights, we start to search from an equally weighted portfolio.

As for θ_6, starting from Eq. (4.2), we can easily obtain a measure for mean and variance of the portfolio, and so we can easily find the expected alpha and the expected Sharpe Ratio for such portfolio. Moreover, we exploit as available criterion also the historical Sharpe Ratio, i.e. that obtained considering the historical mean and variance.

5.2. Hyperparameter optimization and accuracy metrics

Our strategy to determine the hyperparameter vector(s) to use is to exploit the grid-search considering a 7-fold nested Cross Validation (the length of the folds must be equal to θ_4). In particular, our aim is to obtain a subset $\Theta^* \subset \Theta$ made up of the hyperparameters that seem to provide quite stable strategies. To do this, let consider:

- $\varphi : \Theta \to \mathcal{P}$ the function that, starting from the available data and a vector of hyperparameters θ gives us the selected portfolio to invest in. φ is the core of our proposal, i.e. is the function that describes the methodology developed in the previous subsections.
- $\Psi = \Psi(P)$ a vector of measures, i.e. a vector of functions $\psi(P)$ from the universe of portfolios $\mathcal{P}$ to $\mathbb{R}$ such that each function corresponds to a particular measure.
- A vector of survival thresholds.

Then we save those θ such that the corresponding $\Psi(\varphi(\theta))$ exceed these thresholds. In particular, we consider 5 performance measures, so $\Psi = (\psi_1, \dots, \psi_5) : \mathcal{P} \to \mathbb{R}^5$ and we indicate the generic portfolio with $P = \{P_t\}_{t \in \{1, \dots, K\}}$:

Return (P%) : $\psi_1(P) = 100(\frac{P_K}{P_1} - 1)$ One of the more intuitive measures to evaluate trading strategies is the profit achieved by each strategy. We express it in percentage. It takes into account only the profitability of the strategy.

Max Drawdown (MD%) : $\psi_2(P) = 100 \max_{t<s\in T}\{1 - \frac{P_s}{P_t}\}$ It is defined as the biggest difference between a local maximum and the subsequent local minimum (i.e. the maximum loss), expressed in percentage. The MD is a very useful tool to investigate the riskiness of a strategy. For this reason, it is widely used in financial literature (see for example the work of Chekhlov, Uryasev, and Zabarankin (2005)). However, it takes into account only the riskiness of the strategy.

Profit Factor (PF) : $\psi_3(P) = \frac{\sum_{t=2}^{K}(P_t - P_{t-1})^+}{\sum_{t=2}^{K}(P_t - P_{t-1})^-}$ It is defined as the ratio between earned and lost ticks, so it takes into account not only profit but also the loss suffered by the strategy. It is a number, greater or equal to 0, that is less than 1 if the strategy close with a loss, greater than 1 if there is a gain. It gives us an idea of the evolution of the strategy: the more this value is close to 1, the more uncertain is the strategy, with a frequent succession of gains and losses. There are various works that exploit this metric, see for example that of Bakhach, Chinthalapati, Tsang, and El Sayed (2018).

Recovery Factor (RF) : $\psi_4(P) = (P_K - P_1)/\max_{t<s\in T}\{P_t - P_s\}$ It is defined as the ratio between profit and max drawdown, both of them expressed in points. It supplies us an indication of the capability of the reaction of our algorithm and it is employed in various works related to trading and investment strategies, such as that of Kisela, Virdzek, and Vajda (2015).

Sharpe Ratio (ShR) : $\psi_5(P) = \frac{\mu}{\sigma}$ $with$ $\mu = E(\frac{P_t}{P_{t-1}} - 1)$, $\sigma = \sqrt{Var(\frac{P_t}{P_{t-1}} - 1)}$ It was introduced by Sharpe (1975), and it is defined as the ratio between the risk premium, that is the difference between expected return and the risk-free rate, and standard deviation, which is a proxy of the riskiness of the strategy. It provides a measurement of how the riskiness of the strategy is remunerated.

Furthermore, in order to obtain a more stable estimate of the measures used in the 7-fold nested cross validation stage, we evaluate both the performance measures calculated on all the folds that those obtained excluding the outlier values, i.e. removing those values that are out of the interval $[Q_1 - IQ^*, Q_3 + IQ^*]$ where Q_1 and Q_3 are, respectively, the 0.25, 0.75 quantiles and IQ^* is $1.5 * Interquartile\ Range$. Finally, after the selection, we test the elements of Θ^* in subsequent 2 folds, comparing them with some benchmarks.

6. Experimental results

We test our methodology on three different markets, namely Italian Stock Market, Exchange Rates (or Forex) and Cryptocurrencies (or Crypto). Each of them has characterized by different peculiarities. We make three different experiments on these three different datasets, without any interaction with each other. In order to have a reliable estimate of the performance of the market-neutral portfolio, we split the total amount of data into a train set and two test set, whose dimension is equal to the temporal horizon of the strategy. We perform the 7-fold nested cross validation looking exclusively at the train set, and so we obtain the subset Θ^*. Then, we test the vectors in Θ^*. For the first test set, the clustering and the construction of the market-neutral portfolio are done exploiting only the data in the train set, while for the second test we expand the train set including also the first test set. Furthermore, for each market, we provide two examples of hyperparameters in Θ^*: one that obtains a non-negligible profit in the test sets, and the other one that suffers a loss. Moreover, we underline that when we compute the weights we consider only three decimal places, i.e. we neglect contributions lower than 0.1% of total capital. Finally, we compare our strategy with some benchmarks. We exploit the minimum-variance portfolio, the mean–variance portfolio constructed by optimizing the ShR, and whenever it is possible, the Buy & Hold strategy on an appropriate index.

Table 4
Italian Market — Percentage distributions of hyperparameters.

Risk factors extraction		
PCA 25.0	h-NNPCA 33.33	VAE 41.67
Number of PCs		
Five 58.33		Six 41.67
Scaler		
Standard 91.67		MinMax 8.33
Time horizon		
Two 16.67	Three 33.33	Four 50.0
Portfolios' length		
$\lvert\mathcal{A}\rvert + 1 = 66.67$		$\lvert\Omega\rvert = 33.33$
Criterion		
Alpha 33.33	Sharpe 16.67	Expected sharpe 50.0
Sign		
Direct 91.67		Inverse 8.33

6.1. Test - Italian stock market

We collect the time series of the returns of 30 stocks belonging to the Italian market, among those with higher capitalization. The historical data are provided by *mercati.ilsole24ore.com* and the period considered is from 2010-10-26 to 2020-12-31. As for the Buy & Hold strategy, we exploit the **FTSE MIB**, that is the Italian index. Regarding the set of admissible vectors of hyperparameters, we consider the following one with cardinality equal to 432:

$$\begin{aligned}\Theta = &\{PCA, h-NNPCA, VAE\} \times \{5,6\} \times \{Standard, MinMax\} \\ &\times\{2,3,4\} \times \\ &\times\{\theta_5(\mathcal{A},\Omega) = |\mathcal{A}| + 1, \quad \theta_5(\mathcal{A},\Omega) = |\Omega|\} \times \\ &\times\{TotalAlpha, HistoricalSharpe, ExpectedSharpe\} \times \{-1,1\}\end{aligned}$$

To understand the nomenclature used, let us rewrite Θ in this way:

$$\Theta = \{P,N,V\} \times \{5,6\} \times \{S,M\} \times \{D,T,Q\} \times \{n,m\} \times \{A,F,T\} \times \{D,I\}$$

We exploit the following thresholds: **Percentage Profit**: 1.5%; **Profit Factor**: 1.2; **Percentage Drawdown**: 10%; **Recovery Factor**: 1; **Sharpe Ratio**: 1. There are 12 hyperparameters that pass the thresholds. Their characteristics are described in Table 4.

As an example of a losing strategy, we consider the N6SDnTD (following the nomenclature above explained, 6 risk factors extracted via h-NNPCA and Standard scaler, we invest according to those portfolios, made up of the number of principal components + 1 stocks, that maximize the historical Sharpe Ratio). Table 5 shows the results obtained in the two test sets, comparing them with those obtained by the benchmarks (note that **B** is for the Buy & Hold strategy, **MinVar** is for the minimum variance portfolio, **Sharpe** is for the Sharpe optimal portfolio, and **SPs** is for Selected Portfolio).

Fig. 4 shows the performance of the strategy in the sets considered.

As an example of a winning strategy, we exploit N6SQnTD, whose results are shown in Table 6 and in Fig. 5.

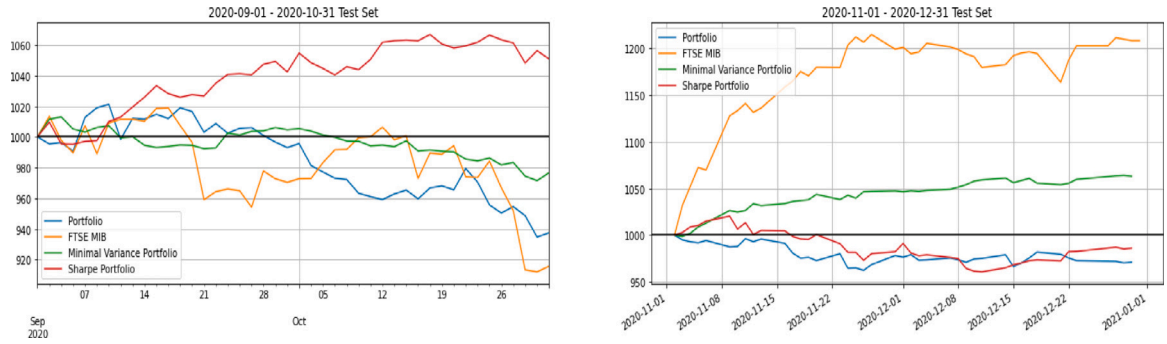

Fig. 4. Comparison between N6SDnTD and the benchmarks - Example 1.

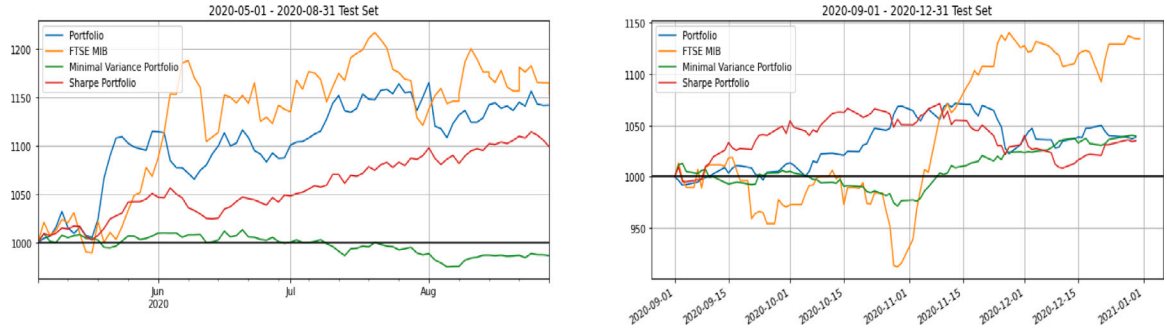

Fig. 5. Comparison between N6SQnTD and the benchmarks - Example 2.

Table 5
Results obtained by N6SDnTD in Italian Market.

		P%	PF	MD%	RF	ShR
	B	−8.43	0.67	10.48	−0.79	−1.66
20-09	**Min Var**	−2.34	0.68	4.1	−0.56	−2.03
20-10	**Sharpe**	**5.08**	**1.64**	**1.74**	**2.74**	**3.53**
	SPs	−6.24	0.62	8.46	−0.72	−2.38
	B	**20.83**	2.94	4.21	**4.07**	**9.23**
20-11	**Min Var**	6.32	**3.27**	**0.68**	0.14	8.31
20-12	**Sharpe**	−1.39	0.86	5.84	3.94	−0.89
	SPs	−2.89	0.7	3.78	−0.77	−1.93

Table 6
Results obtained by N6SQnTD in Italian Market.

		P%	PF	MD%	RF	ShR
	B	**15.25**	1.31	7.87	1.59	1.98
20-05	**Min Var**	−1.36	0.89	3.79	−0.36	−0.69
20-08	**Sharpe**	9.86	**1.7**	**3.01**	**3.1**	**4.02**
	SPs	14.16	1.47	4.88	2.49	2.77
	B	**13.46**	1.38	10.48	**1.26**	1.76
20-09	**Min Var**	3.97	**1.4**	**4.1**	0.96	**2.0**
20-12	**Sharpe**	3.51	1.19	5.87	0.56	1.16
	SPs	3.88	1.23	4.59	0.79	1.36

6.2. Test - Exchange rates

For the Forex data, we have a universe of 26 time series referred to various exchange rates, both majors such as EUR/GBP and minors like EUR/AUD. All data are provided by ActivTrades broker, are daily and referred to the CFD, and cover the period between 2010-01-04 and 2020-12-31. The set of admissible vectors of hyperparameters Θ is the same as the one above. The thresholds imposed are the following: **Percentage Profit**: 0.6%; **Profit Factor**: 1.2; **Percentage Drawdown**: 5%; **Recovery Factor**: 1; **Sharpe Ratio**: 1. Also in this case, 12 hyperparameters pass the survival thresholds, and their properties are described in Table 7.

Table 7
Forex — Percentage distributions of hyperparameters.

Risk factors extraction		
PCA 41.67	h-NNPCA 0	VAE 58.33
Number of PCs		
Five 33.33		Six 66.67
Scaler		
Standard 41.67		MinMax 58.33
Time horizon		
Two 33.33	Three 50.0	Four 16.67
Portfolios' length		
$\lvert\mathcal{A}\rvert + 1$ = 66.67		$\lvert\Omega\rvert$ = 33.33
Criterion		
Alpha 41.67	Sharpe 33.33	Expected sharpe 25.0
Sign		
Direct 16.67		Inverse 83.33

As an example of a losing strategy, we propose V6MTnTI. The performances of the strategy in the two test sets are summarized in Table 8 and Fig. 6.

As for the winning strategy, the example is V6MDmAI and its performances are reported in Table 9 and are shown in Fig. 7.

6.3. Test - Cryptocurrencies

The data used for Crypto are referred to the exchange rates of some cryptos against the Tether (symbol USDT), which can be considered as a proxy of 1 dollar (according to the official Tether website "Tether converts cash into digital currency, to anchor or tether the value to the price of national currencies like the US dollar"). We work with a universe of 24 cryptos. The data are provided by Invevsting.com and they are referred to the period between 2018-03-23 and 2020-12-31. Because of this lack of data, when we consider the quarter time

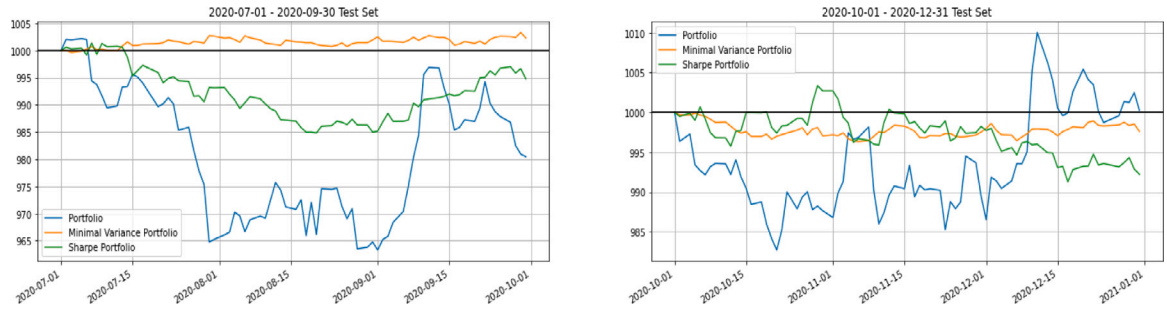

Fig. 6. Comparison between V6MTnTI and the benchmarks — Example 1.

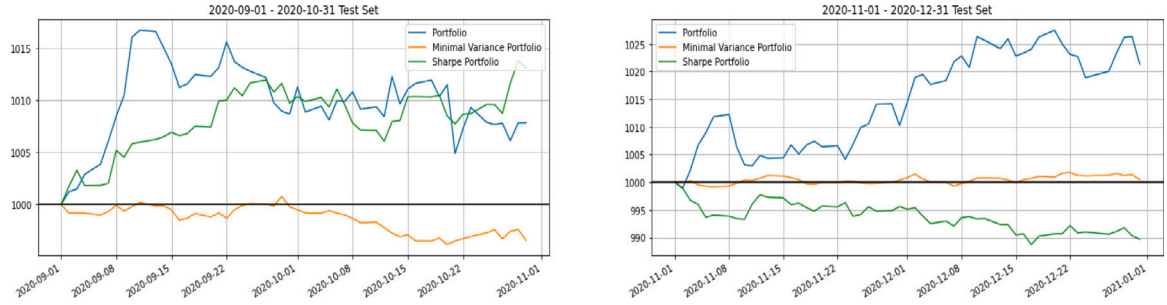

Fig. 7. Comparison between V6MDmAI and the benchmarks - Example 2.

Table 8
Results obtained by V6MTnTI in Forex market.

		P%	PF	MD%	RF	ShR
20-07/20-09	**MinVar**	**0.23**	**1.17**	**0.2**	**1.17**	**0.99**
	Sharpe	−0.52	0.86	1.65	−0.32	−0.88
	SPs	−1.95	0.82	3.88	−0.5	−1.09
20-10/20-12	MinVar	−0.24	0.84	**0.36**	−0.66	−1.15
	Sharpe	−0.78	0.8	1.2	−0.64	−1.35
	SPs	**0.03**	**1.0**	1.73	**0.02**	**0.02**

Table 9
Results obtained by V6MDmAI in Forex market.

		P%	PF	MD%	RF	ShR
20-09/20-10	**MinVar**	−0.34	0.71	**0.46**	−0.74	−2.2
	Sharpe	**1.31**	**1.77**	0.58	**2.23**	**3.45**
	SPs	0.78	1.24	1.16	0.66	1.27
20-11/20-12	**MinVar**	0.04	1.05	**0.22**	0.21	0.33
	Sharpe	−1.03	0.62	1.13	−0.91	−2.8
	SPs	**2.13**	**1.5**	0.92	**2.3**	**2.7**

Table 10
Crypto — Percentage distributions of hyperparameters.

Risk factors extraction		
PCA 35.71	h-NNPCA 57.14	VAE 7.14
Number of PCs		
Four 14.29	Five 14.29	Six 71.43
Scaler		
Standard 57.14		MinMax 42.86
Time horizon		
One 100.0	Two 0.0	Three 0.0
Portfolios' length		
$\|\mathcal{A}\|+1 = 71.43$		$\|\Omega\| = 28.57$
Criterion		
Alpha 35.71	Sharpe 28.57	Expected sharpe 35.71
Sign		
Direct 100.0		Inverse 0.0

horizon, we have to work with only 5 (and not 7) folds. As for the Buy & Hold strategy, we exploit an index weighted by the log of the market cap of each crypto (we use the log to contain the predominance of Bitcoin). Regarding the set of admissible vectors of hyperparameters, we consider the following one made up of 648 elements:

$$\begin{aligned}\Theta = &\{PCA, h-NNPCA, VAE\}\times\{4,5,6\}\times\{Standard, MinMax\}\\ &\times\{1,2,3\}\\ &\times\{\theta_5(\mathcal{A},\Omega)=|\mathcal{A}|+1,\quad \theta_5(\mathcal{A},\Omega)=|\Omega|\}\\ &\times\{TotalAlpha, HistoricalSharpe, ExpectedSharpe\}\times\{-1,1\}\end{aligned}$$

To understand the nomenclature used, let us rewrite Θ in this way:

$$\Theta = \{P,N,V\}\times\{4,5,6\}\times\{S,M\}\times\{U,D,T\}\times\{n,m\}\times\{A,F,T\}\times\{D,I\}$$

We use these thresholds: **Percentage Profit**: 1.5%; **Profit Factor**: 1.2; **Percentage Drawdown**: 8%; **Recovery Factor**: 1; **Sharpe Ratio**: 1. The distribution of the 14 hyperparameters vectors that pass the threshold is reported in Table 10.

The example of the losing strategy is P6MUnTD and its results on the tests are summarized in Table 11 and Fig. 8.

Regarding the winning strategy, it is N6SUnTD. Table 12 and Fig. 9 show the results obtained by the strategy during the test sets.

6.4. Discussion

Regarding the results obtained in the construction of Θ^*, it is very interesting to observe the situation concerning the sign: in fact, while for Forex we invest almost always against the optimal portfolios, in the stock and crypto markets we invest in agreement with the selected portfolios. Clearly, there are big differences between these markets (for example the liquidity, the volume exchanged but also the volatility) and it is quite interesting to observe how these structural differences lead to different investment strategies. Furthermore, it can be worthwhile trying to understand the causes of this effect, maybe exploring other markets.

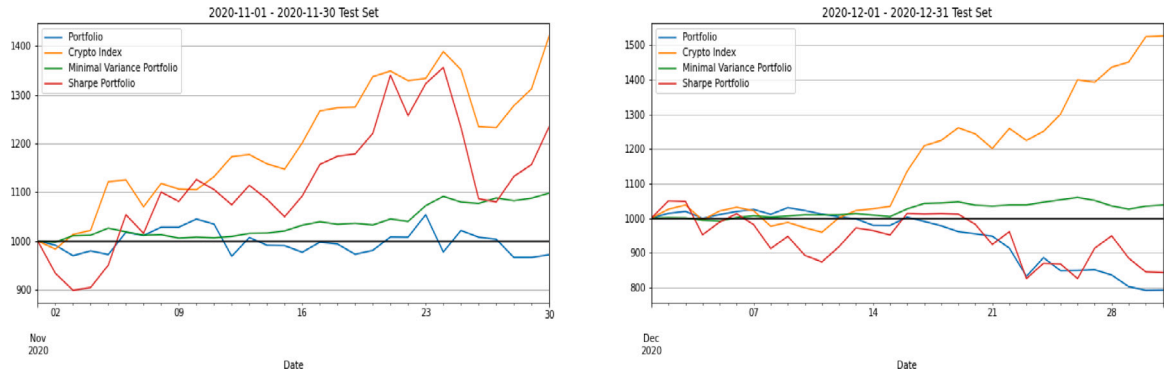

Fig. 8. Comparison between P6MUnTD and the Crypto benchmarks — Example 1.

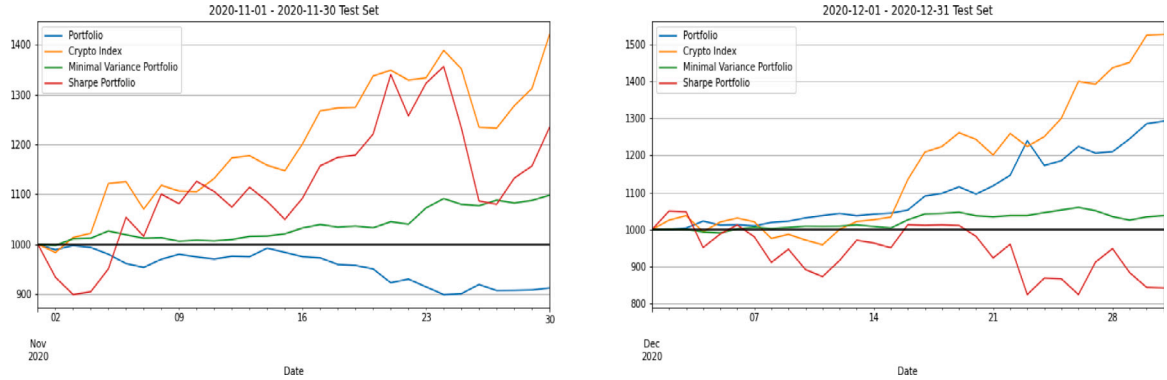

Fig. 9. Comparison between N6SUnTD and the Crypto benchmarks — Example 2.

Table 11
Results obtained by P6MUnTD in Crypto market.

		P%	PF	MD%	RF	ShR
	B	**42.03**	2.45	11.23	2.7	**30.81**
20-11	**MinVar**	9.83	**2.66**	**1.99**	**4.81**	8.39
20-11	**Sharpe**	23.44	1.37	20.35	0.85	5.64
	SPs	−2.79	0.91	8.32	−0.32	−0.48
	B	**52.61**	**3.29**	7.58	**6.69**	**57.35**
20-12	**MinVar**	3.79	1.55	**3.21**	1.11	2.97
20-12	**Sharpe**	−15.73	0.77	21.43	−0.7	−0.89
	SPs	−20.79	0.41	23.17	−0.87	−2.12

Table 12
Results obtained by N6SUnTD in Crypto market.

		P%	PF	MD%	RF	ShR
	B	**42.03**	2.45	11.23	2.7	**30.81**
20-11	**MinVar**	9.83	**2.66**	**1.99**	**4.81**	8.39
20-11	**Sharpe**	23.44	1.37	20.35	0.85	5.64
	SPs	−8.76	0.51	10.09	−0.87	−2.94
	B	**52.61**	3.29	7.58	**6.69**	**57.35**
20-12	**MinVar**	3.79	1.55	**3.21**	1.11	2.97
20-12	**Sharpe**	−15.73	0.77	21.43	−0.7	−0.89
	SPs	29.25	**3.33**	5.41	4.36	20.28

Regarding the other hyperparameters, another interesting situation is that regarding the scaler used: in fact, while in the Forex and crypto markets there is a situation of substantial equivalence between the Standard and the MinMax one, in the stocks market there is a preponderance of the first one, that we remember is the one that verifies the hypothesis of the APT.

Moreover, we observe that for the cryptocurrencies only strategies with a temporal horizon of 1 month survive the filter. Also in this case, it may be interesting to investigate the causes behind this phenomenon (for example if the lack of a long series of data is a problem, or if the high volatility affects the efficiency on a long time horizon).

Furthermore, there are some general conclusions that we can infer. Firstly, we observe that the PCA is a good choice in almost all situations. In fact, it seems to be able to provide good risk factors that are well-suited for our strategy, and thus we find that many winning strategies exploit the PCA. Also, it seems that more PCs is better, in the sense that often, the winning strategies are those that exploit more PCs. This could suggest the effectiveness of the features selection and clustering stages. Moreover, it also seems that the Sharpe Ratio criterion, which exploits historical data, has lower importance than the ones that exploited expected values in identifying the winning strategies.

Finally, regarding the results obtained in evaluating the strategies of Θ^* in the test sets, we provide both a losing and a winning example. We do this to emphasize that the first selection stage with the survival thresholds alone is not able to identify only the winning strategies.

7. Conclusion

The first results in this study provide some ideas that could be useful in assets clustering and/or in the construction of a market-neutral portfolio. However, this idea is still in an embryonic stage, so there are many directions that can be taken in order to improve the performances.

Firstly, as we already pointed out, the cross validation itself is not able to discard all the losing strategies. So, we can make an attempt to resolve this problem by exploiting the test sets to further filter the strategies and, after this last filter, try to use the best ones in a demo test (that is a test with a demo account on a real broker). Furthermore, we can also try to further investigate the model residual, i.e. the idiosyncratic risks of the stocks, to found other hidden patterns which our methodology, and in particular the use of the Sharpe Ratio, is not able to discover.

Then we could change the time-granularity considered, for example switching to weekly data or to intraday data. In particular, this second way is, in our opinion, very interesting because it could be the starting point for some medium-frequency trading strategies, with exit conditions determined by Take Profit and Stop Loss, designed to make the positions opened between some hours and few days. Regarding the time-granularity, there is also another interesting extension that could be considered. More in detail, we can try to extract different sets of risk factors aggregating data into upper granularity and then apply the feature extraction technique to obtain long-term risk factors. For example, starting from daily observations, we can obtain a short-term risk set extracting the risk factors from daily data (as we have already done in this work), then we can aggregate daily data into monthly ones, and then extract another set of features. We expect the utility of this strategy, especially when the investment time horizon is significantly higher than the data time-granularity. However, we underline that there is a big drawback to keep in mind: the introduction of another set of risk factors could generate a collinearity problem, in that no one guarantees that risk factors belonging to different sets are still independent of each other.

Furthermore, we can try to change the features selection (for example we can use Adaptive Elastic Net) to address the collinearity problem and allow a more flexible features extraction stage.

Moreover, about the investment methodology, we can try to extend the hyperparameters space or we can try to change the assets universe, for example choosing stocks belonging to different markets and study them all together. Moreover, in order to produce a strategy that can be really applied to the markets, we have to take into account also practical problems that we have overlooked, such as transaction costs or the divisibility of the assets.

Finally, there are other ideas that could have a more wide horizon. We can try to generalize the problem to data different from financial ones, or we can try to understand how the number n of principal components chosen, or the number of principal components admitted in the cross validation stage, can impact the resulting number of clusters.

CRediT authorship contribution statement

Salvatore Cuomo: Conceptualization, Methodology, Investigation. **Federico Gatta:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft. **Fabio Giampaolo:** Data curation, Formal analysis. **Carmela Iorio:** Methodology, Investigation, Writing – review & editing. **Francesco Piccialli:** Conceptualization, Methodology, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. List of notations

- A-LASSO → Adaptive Least Absolute Shrinkage and Selection Operator
- APT → Arbitrage Pricing Theory
- B → Buy & Hold
- h-NNPCA → hierarchical Neural Network Principal Component Analysis
- LASSO → Least Absolute Shrinkage and Selection Operator
- MD% → Max Drawdown
- MinVar → a Minimum Variance Portfolio
- MSE → Mean Square Error
- NN → Neural Networks
- NNPCA → Neural Network Principal Component Analysis
- OLS → Ordinary Least Squares
- P% → Return
- PCA → Principal Component Analysis
- PCs → Principal Components
- PF → Profit Factor
- RF → Recovery Factor
- Sharpe → Sharpe Optimal Portfolio
- ShR → Sharpe Ratio
- SLSQP → Sequential Least Squares Programming
- SPs → Selected Portfolio
- VAE → Variational AutoEncoders

References

Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering–a decade review. *Information Systems*, *53*, 16–38.

Aghabozorgi, S., & Teh, Y. W. (2014). Stock market co-movement assessment using a three-phase clustering method. *Expert Systems with Applications*, *41*(4), 1301–1314.

Alonso, A. M., & Maharaj, E. A. (2006). Comparison of time series using subsampling. *Computational Statistics & Data Analysis*, *50*(10), 2589–2599.

Avellaneda, M., & Lee, J.-H. (2010). Statistical arbitrage in the US equities market. *Quantitative Finance*, *10*(7), 761–782.

Bakhach, A., Chinthalapati, V. L. R., Tsang, E. P. K., & El Sayed, A. R. (2018). Intelligent dynamic backlash agent: A trading strategy based on the directional change framework. *Algorithms*, *11*(11).

Bilal, M., Ullah, M., & Ullah, H. (2019). Chemometric data analysis with autoencoder neural network. *Electronic Imaging*, *2019*(1), 679–1.

Blitz, D., Huij, J., & Martens, M. (2011). Residual momentum. *Journal of Empirical Finance*, *18*(3), 506–521.

Chatterjee, S., & Hadi, A. S. (2015). *Regression Analysis by Example*. John Wiley & Sons.

Chekhlov, A., Uryasev, S., & Zabarankin, M. (2005). Drawdown measure in portfolio optimization. *International Journal of Theoretical and Applied Finance*, *8*(01), 13–58.

Chen, N.-F., Roll, R., & Ross, S. A. (1986). Economic forces and the stock market. *Journal of Business*, *59*(3), 383–403.

Chiu, K.-C., & Xu, L. (2004). Arbitrage pricing theory-based Gaussian temporal factor analysis for adaptive portfolio management. *Decision Support Systems*, *37*(4), 485–500, Data mining for financial decision making.

Clare, A. D., & Thomas, S. H. (1994). Macroeconomic factors, the APT and the UK stockmarket. *Journal of Business Finance & Accounting*, *21*(3), 309–330.

Corduas, M., & Piccolo, D. (2008). Time series clustering and classification by the autoregressive metric. *Computational Statistics & Data Analysis*, *52*(4), 1860–1872.

De Luca, G., & Zuccolotto, P. (2011). A tail dependence-based dissimilarity measure for financial time series clustering. *Advances in Data Analysis and Classification*, *5*(4), 323–340.

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*(1), 3–56.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, *96*(456), 1348–1360.

Fu, T.-c., Chung, F.-l., Ng, V., & Luk, R. (2001). Pattern discovery from stock time series using self-organizing maps. In *Workshop Notes of KDD2001 Workshop on Temporal Data Mining, Vol. 1*. Citeseer.

de Guevara Cortés, R. L., & Porras, S. T. (2014). Estimation of the underlying structure of systematic risk with the use of principal component analysis and factor analysis. *Contaduría y Administración*, *59*(3), 197–234.

Ladrón de Guevara Cortés, R., Torra Porras, S., & Monte Moreno, E. (2018). Extraction of the underlying structure of systematic risk from non-Gaussian multivariate financial time series using independent component analysis: Evidence from the mexican stock exchange. *Computación y Sistemas*, *22*(4), 1049–1064.

Ladrón de Guevara Cortés, R., Torra Porras, S., & Monte Moreno, E. (2019). Neural networks principal component analysis for estimating the generative multifactor model of returns under a statistical approach to the arbitrage pricing theory. Evidence from the mexican stock exchange. *Computación y Sistemas*, *23*(2), 281–298.

Imajo, K., Minami, K., Ito, K., & Nakagawa, K. (2020). Deep portfolio optimization via distributional prediction of residual factors. arXiv preprint arXiv:2012.07245.

Imajo, K., Minami, K., Ito, K., & Nakagawa, K. (2021). Deep Portfolio Optimization via Distributional Prediction of Residual Factors. In *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35*, (1), (pp. 213–222).

Iorio, C., Frasso, G., D'Ambrosio, A., & Siciliano, R. (2016). Parsimonious time series clustering using p-splines. *Expert Systems with Applications*, *52*, 26–38.

Iorio, C., Frasso, G., D'Ambrosio, A., & Siciliano, R. (2018). A P-spline based clustering approach for portfolio selection. *Expert Systems with Applications*, *95*, 88–103.

Kim, J. H. (2019). Multicollinearity and misleading statistical results. *Korean Journal of Anesthesiology*, *72*(6), 558.

Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392.

Kisela, P., Virdzek, T., & Vajda, V. (2015). Trading the equity curves. *Procedia Economics and Finance*, *32*, 50–55.
Košmelj, K., & Batagelj, V. (1990). Cross-sectional approach for clustering time varying data. *Journal of Classification*, *7*(1), 99–109.
Lai, C.-P., Chung, P.-C., & Tseng, V. S. (2010). A novel two-level clustering method for time series data analysis. *Expert Systems with Applications*, *37*(9), 6319–6326.
Liao, T. W. (2005). Clustering of time series data-a survey. *Pattern Recognition*, *38*(11), 1857–1874.
Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*, *47*(1), 13–37.
Maharaj, E. A., D'urso, P., & Galagedera, D. U. (2010). Wavelet-based fuzzy clustering of time series. *Journal of Classification*, *27*(2).
Mancisidor, R. A., Kampffmeyer, M., Aas, K., & Jenssen, R. (2021). Learning latent representations of bank customers with the variational autoencoder. *Expert Systems with Applications*, *164*, Article 114020.
Meinshausen, N., Bühlmann, P., et al. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, *34*(3), 1436–1462.
Montesdeoca, L., Squires, S., & Niranjan, M. (2019). Variational autoencoder for non-negative matrix factorization with exogenous inputs applied to financial data modelling. In *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)* (pp. 312–317). IEEE.
Mossin, J. (1966). Equilibrium in a capital asset market. *Econometrica*, 768–783.
Nair, B. B., Kumar, P. S., Sakthivel, N., & Vipin, U. (2017). Clustering stock price time series data to generate stock trading recommendations: An empirical study. *Expert Systems with Applications*, *70*, 20–36.
Nakagawa, K., Kawahara, T., & Ito, A. (2020). Asset allocation strategy with non-hierarchical clustering risk parity portfolio. *Journal of Mathematical Finance*, *10*(4), 513–524.
Nicholas, J. G. (2000). *Market Neutral Investing*. Bloomberg Press Princeton, NJ.
Omran, M. F. (2005). Identifying risk factors within the arbitrage pricing theory in the Egyptian stock market. *University of Sharjah Journal*, *2*(2), 103–119.
Otranto, E. (2008). Clustering heteroskedastic time series by model-based procedures. *Computational Statistics & Data Analysis*, *52*(10), 4685–4698.
Panagiotidis, T., Stengos, T., & Vravosinos, O. (2018). On the determinants of bitcoin returns: A LASSO approach. *Finance Research Letters*, *27*, 235–240.
Piccolo, D. (1990). A distance measure for classifying ARIMA models. *Journal of Time Series Analysis*, *11*(2), 153–164.
Pole, A. (2011). *Statistical Arbitrage: Algorithmic Trading Insights and Techniques, Vol. 411*. John Wiley & Sons.
Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, *13*(3), 341–360.
Ross, S. (1977). Risk, return and arbitrage. *Risk and Return in Finance*, *1*, 189–218.
Scholz, M. (2006). Nonlinear PCA toolbox for matlab. URL http://www.nlpca.org/matlab.html.
Scholz, M., Fraunholz, M., & Selbig, J. (2008). Nonlinear principal component analysis: neural network models and applications. In A. N. Gorban, B. Kégl, D. C. Wunsch, & A. Zinovyev (Eds.), *LNCSE: vol. 58, Principal Manifolds for Data Visualization and Dimension Reduction* (pp. 44–67). Springer.
Scholz, M., & Vigário, R. (2002). Nonlinear PCA: a new hierarchical approach. In M. Verleysen (Eds.), *Proceedings ESANN*, (pp. 439–444).
Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, *19*(3), 425–442.
Sharpe, W. F. (1975). Adjusting for risk in portfolio performance measurement. *The Journal of Portfolio Management*, *1*(2), 29–34.
Sharpe, W. F., Alexander, G. J., & Bailey, J. W. (1998). *Investments* (sixth edn). Upple Saddle River, NJ: Prentice-Hall.
Spyridis, T., Sević, & Theriou, N. (2012). Macroeconomic vs. Statistical APT approach in the athens stock exchange. *International Journal of Business*, *17*, 39.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.
Tzagkarakis, G., Caicedo-Llano, J., & Dionysopoulos, T. (2013). Exploiting market integration for pure alpha investments via probabilistic principal factors analysis. *Journal of Mathematical Finance*, *3*(1A), 192–200.
Van Wijk, J. J., & Van Selow, E. R. (1999). Cluster and calendar based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)* (pp. 4–9). IEEE.
Visagie, G., & Hoffman, A. (2017). Comparison of statistical arbitrage in developed and emerging markets. *International Journal of Trade, Economics and Finance*, *8*(2), 67–72.
Yip, F., & Xu, L. (2000). An application of independent component analysis in the arbitrage pricing theory. In *Proceedings of the IEEE-INNS-ENNS INternational Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Vol. 5*, (pp. 279–284).
Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, *101*(476), 1418–1429.
Zou, H., & Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. *The Annals of Statistics*, *37*(4), 1733.