

On Prediction of Stock Return and Volatility Using Clustering Techniques: Taking an Example of Japanese Stock Market

Jieni Liu*, Hisashi Tanizaki

Graduate School of Economics, The University of Osaka, Osaka, Japan
Email: *pngjieni@gmail.com, tanizaki@econ.osaka-u.ac.jp

How to cite this paper: Liu, J., & Tanizaki, H. (2025). On Prediction of Stock Return and Volatility Using Clustering Techniques: Taking an Example of Japanese Stock Market. *Open Journal of Social Sciences*, 13, 532-545.
<https://doi.org/10.4236/jss.2025.1310031>

Received: September 15, 2025

Accepted: October 21, 2025

Published: October 24, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Stock returns exhibit nonlinear dynamics and volatility clustering. It is well known that we cannot forecast the movements of stock prices under the condition that market is efficient. In most research, it is concluded that stock markets are efficient and accordingly stock returns are not predictable. However, using some clustering techniques and choosing the stock returns in the cluster giving us high returns, in this paper we examine whether stock returns are predictable or not. To address this issue, we combine various data preprocessing techniques with three clustering methods (i.e., one K-means and two K-medoids clustering algorithms) in Japanese Nikkei 225 financial market. As a result, we cannot predict stock price returns, but we can predict volatility of stock returns. This result is consistent with a lot of past studies.

Keywords

Stock Return, Volatility, Clustering, K-Means, K-Medoids

1. Introduction

Empirically, stock returns display significant nonlinear dynamics and stochastic volatility. Phenomena such as volatility persistency pose significant challenges to traditional statistical methods. Time series clustering has emerged as an effective tool for identifying structural similarities among stock returns, offering investors deeper insights into market structures. A comprehensive review by Liao (2005) systematically examines various clustering techniques, addressing crucial aspects including data representation, similarity measures, clustering algorithms, and clustering validation criteria. The importance of adapting these techniques to specific analytical contexts is particularly emphasized. In financial research, cluster-

ing methods are widely used to uncover underlying market structures and classify assets based on their return characteristics. Early foundational work by Mantegna (1999) introduces the minimum spanning trees to identify hierarchical correlation structures among stocks, laying the theoretical groundwork for analyzing market correlation structures. Subsequent studies expanded on this framework with different practical emphasis.

For example, Tola et al. (2008) utilize hierarchical clustering explicitly for portfolio construction, demonstrating enhanced investment performance. Tumminello et al. (2010) analyze asset correlations to improve portfolio diversification and risk management. Brida and Risso (2009) examine structural relationships among the returns of major North American companies. However, these studies typically used datasets limited to specific markets, thus restricting the generalizability and robustness of their conclusions.

Recent empirical studies investigated practical applications of clustering methods. Chen et al. (2021) compare various clustering algorithms using extensive data from China's A-share market. They conclude that clustering-based classifications could effectively replace traditional industry classifications and enhance portfolio performance.

Marti et al. (2016) investigate how varying time window lengths affect clustering results, highlighting substantial impacts on both clustering performance and interpretability. The clustering methods such as K-means and K-medoids minimize sum of the distances between sample data and their clustered sample mean, where we often utilize sum of squared differences between sample data and their sample mean, so-called the Euclidean distance or L2 distance. Shi and Xiao (2022) integrate the dynamic time warping distance (hereafter, DTW) into the K-means algorithm, enhancing its capability to capture temporal shifts in financial data. Paparrizos and Gravano (2015) develop the k-shape method utilizing the shape-based distance (hereafter, SBD), employing normalized cross-correlation as a distance metric to improve clustering stability. Thus, not only the Euclidean distance but also DTW and SBD are investigated as the distance metrics. López-Oriona et al. (2025) propose a forecast error-based distance metric, emphasizing the alignment of distance metrics with specific analytical objectives. Paparrizos et al. (2024) observe that many studies evaluate clustering techniques in isolation. They recommend integrating traditional clustering methods with deep learning approaches and evaluating all methods using consistent criteria for fair comparison. Similarly, Drago (2024) proposes to ensemble clustering strategies to improve robustness, emphasizing the need for systematic method comparisons.

While a lot of existing studies have significant strides, few studies have systematically analyzed how combinations of data preprocessing techniques and distance metrics impact clustering results, particularly across varied market environments. To address these problems, this paper systematically investigates the interplay between data preprocessing techniques and distance metrics, evaluating their joint impact on both clustering results and clustering-based portfolio construction.

Specifically, this paper considers seven data preprocessing techniques (i.e., simple returns, Mean-Min-Max, Min-Max, Z-score, L1-norm, L2-norm, and robust scaling shown in Section 2) combined with three distance metrics (i.e., Euclidean distance in K-means clustering method, and DTW and SDB in the K-medoids clustering method in Section 3). The combinations between the data preprocessing techniques and the clustering methods are compared with respect to their effectiveness and common usage in financial research. We implement these methods on the stock return data constructing Japanese Nikkei 225 index. The empirical findings illustrate the impact of methodological choices on clustering results, offering practical insights for investors and researchers interested in reliable analytical methods. Through comparative analysis across various market environments, this study contributes to a deeper understanding of the robustness and practical use of clustering methods in finance.

It is well known that stock market is efficient, which implies that we cannot forecast stock returns. In contrast, applying machine learning techniques, [Chun et al. \(2025\)](#) and [Zhang et al. \(2024\)](#) discuss volatility forecasting, which indicates that we can forecast volatilities. [Tan et al. \(2024\)](#) also mention about prediction of volatilities. In this paper, we find that we cannot predict stock price returns but we can predict volatility of stock returns. These results are consistent with a lot of past studies.

The remainder of this paper proceeds as follows. Section 2 introduces data preprocessing techniques, which are applied to stock daily returns. Section 3 discusses clustering algorithms, distance metrics, and evaluation of clustering. Section 4 describes the datasets, experimental design, and the empirical results. Finally, Section 5 gives us the summary and concluding remarks.

2. Data Preprocessing Techniques

Data preprocessing helps reduce scale differences among financial time series, mitigate the impact of outliers, and enhance the discernibility of temporal patterns, such as “shape”. This section presents seven data preprocessing techniques to enhance clustering effectiveness in time series analysis. Each data preprocessing technique is described as follows.

1) Simple Returns

We utilize daily stock closing price in a trading day. P_{it} is defined as the i th stock price at time t . The raw return r_{it} is calculated directly from daily stock price, preserving the original economic meaning of the data as follows:

$$r_{it} = 100 \times \frac{P_{it} - P_{t-1,i}}{P_{t-1,i}} \quad (1)$$

for $t = 1, 2, \dots, T$ and $i = 1, 2, \dots, n$. This transformation is intuitive and retains the direct financial interpretation of returns. However, it fails to mitigate scaling issues and outlier effects, while also lacking efficacy in enhancing temporal pattern discernibility.

2) Mean-Min-Max Normalization

Let $r_i = (r_{i1}, \dots, r_{it}, \dots, r_{iT})'$ denote an $T \times 1$ vector representing a generic time series of the i th stock returns.

\bar{r}_i indicates the sample mean of the i th stock returns for $t = 1, 2, \dots, T$, i.e., $\bar{r}_i = (1/T) \sum_{t=1}^T r_{it}$. $\max(r_i)$ and $\min(r_i)$ denote the maximum and minimum values in the T elements in vector r_i , respectively. Using the r_{it} defined in Equation (1), Mean-Min-Max normalization can be computed as:

$$\frac{r_{it} - \bar{r}_i}{\max(r_i) - \min(r_i)} \quad (2)$$

for $t = 1, 2, \dots, T$ and $i = 1, 2, \dots, n$.

This method unifies the scale of different time series while preserving their relative trends. However, it remains sensitive to outliers, as extreme values can significantly affect \bar{r}_i and expand $\max(r_i) - \min(r_i)$ then destabilize comparisons between time series.

3) Min-Max Normalization

Using the same notation defined previously, Min-Max normalization scales the data solely based on its minimum and maximum values, constraining the normalized values within the interval $[0, 1]$.

$$\frac{r_{it} - \min(r_i)}{\max(r_i) - \min(r_i)} \quad (3)$$

for $t = 1, 2, \dots, T$ and $i = 1, 2, \dots, n$.

This method facilitates direct comparisons of volatility ranges across distinct time series but sacrifices information regarding absolute return levels. Compared to Mean-Min-Max normalization, this method more explicitly highlights differences in the magnitude of fluctuations.

4) Z-Score Normalization

Z-score normalization (also known as zero-mean standardization) scales the data with mean zero and variance one.

$$\frac{r_{it} - \bar{r}_i}{s_i} \quad (4)$$

for $t = 1, 2, \dots, T$ and $i = 1, 2, \dots, n$, where s_i denotes the sample standard deviation of r_i .

This method produces a dimensionless series with zero mean and unit standard deviation, making direct comparisons across different series more straightforward. Although it is widely used in statistical analysis, this method is sensitive to outliers as extreme values can distort the mean and standard deviation values, potentially compromising the reliability of subsequent analysis.

5) L1 Normalization

L1 normalization (also known as Norm1) scales a time series by its L1 norm. Norm1 is denoted as:

$$\frac{r_{it}}{\|r_i\|_1} \quad (5)$$

for $t = 1, 2, \dots, T$ and $i = 1, 2, \dots, n$, where $\|r_i\|_1$ is defined as $\sum_{t=1}^T |r_{it}|$.

The Norm1 highlights the relative scale and distribution of fluctuations within a time series by emphasizing proportional changes. However, it may be less effective in scenarios where distinguishing differences in absolute volatility across time series is crucial.

6) L2 Normalization

L2 normalization (also known as Norm2) scales a time series to have a unit length in terms of its L2 norm. Norm2 is calculated as:

$$\frac{r_{it}}{\|r_i\|_2} \tag{6}$$

for $t = 1, 2, \dots, T$ and $i = 1, 2, \dots, n$, where $\|r_i\|_2 = \sqrt{\sum_{t=1}^T r_{it}^2}$, which emphasizes the direction of trends rather than their magnitude.

The Norm2 is particularly suitable for identifying similarities in the directional trends of time series while disregarding differences in their overall magnitude. Compared to Norm1, Norm2 puts more weight on larger deviations, thereby emphasizing dominant trend patterns rather than proportional fluctuations.

7) Robust Scaling

Robust scaling reduces the influence of outliers by using robust statistics to scale data. Robust scaling can be calculated as:

$$\frac{r_{it} - \text{median}(r_i)}{Q3_i - Q1_i} \tag{7}$$

for $t = 1, 2, \dots, T$ and $i = 1, 2, \dots, n$, where $\text{median}(r_i)$ denotes the median of r_{it} , $t = 1, 2, \dots, T$ in vector r_i , $Q1_i$ and $Q3_i$ represent the first quartile (i.e., 25 percentile) and the third quartile (i.e., 75 percentile) of r_i , respectively. Thus, $Q3_i - Q1_i$ is the interquartile range of the i th stock price returns.

Robust scaling is particularly suitable for financial returns characterized by frequent sharp fluctuations or occasional extreme spikes, as it significantly reduces the influence of outliers. However, because this method emphasizes the central portion of data distribution, it can reduce sensitivity to information in the tails, potentially affecting analysis that rely heavily on tail behavior.

3. Clustering Analysis Methods

This section presents the cluster analysis methods used in this paper. Clustering enables grouping stocks with similar characteristics and provides a foundation for further analysis.

3.1. Clustering Algorithms

This study focuses on two partition-based clustering algorithms, i.e., K-means and K-medoids. These algorithms determine cluster centers differently, but both share the common objective of minimizing dissimilarity within clusters.

1) K-means

The K-means algorithm, introduced by MacQueen (1967), groups data points

by minimizing squared distances between data points and their cluster centers. It starts by choosing K random points as initial cluster centers. Then it repeats two steps: each data point is assigned to the nearest center and each cluster center is recomputed as the mean of all data points assigned to it. These steps are repeated until all cluster centers no longer change or a maximum iteration limit is reached.

The K-means algorithm is given by solving the following problem:

$$\min_{\mu_1, \mu_2, \dots, \mu_K} \sum_{k=1}^K \sum_{r_i \in C_k} \|r_i - \mu_k\|_2 \quad (8)$$

where K is the number of clusters, r_i is the time series data vector (i.e., $T \times 1$ vector), C_k is the set of points in cluster k , and μ_k is the center vector ($T \times 1$ vector) of cluster k .

The K-means algorithm is efficient, simple, and widely applicable.

However, it is sensitive to initial cluster center selection and can be negatively affected by outliers. The method of updating cluster centers by averaging intra-cluster data points in K-means is typically valid under Euclidean distance. When the other distance metrics are applied, the cluster center update mechanism based on arithmetic averaging may cause divergence.

2) K-medoids

Due to the sensitivity of K-means to outliers and distance metrics, alternative robust methods like K-medoids have been developed. Introduced by [Kaufman and Rousseeuw \(1987\)](#), K-medoids selects actual data points as cluster centers instead of means. Initially, it randomly selects K data points as cluster centers. Each data point is then assigned to the closest cluster centers. Subsequently, for each cluster, the algorithm selects a new cluster center that minimizes the total distance from the cluster centers to other cluster points. This iterative process continues until the cluster centers stabilize or the maximum iteration threshold is reached.

The K-medoids algorithm solves the following minimization problem:

$$\min_{\mu_1, \mu_2, \dots, \mu_K} \sum_{k=1}^K \sum_{r_i \in C_k} D(r_i, \mu_k) \quad (9)$$

where μ_k denotes the center of the k th cluster, which is different from the μ_k in K-means, and $D(r_i, \mu_k)$ indicates the distance metrics between two vectors r_i and μ_k . Appropriate distance metrics allow us to accurately capture similarities between time series, which is essential for constructing effective portfolios. In this paper, we consider $D_{DTW}(\cdot, \cdot)$ and $D_{SBD}(\cdot, \cdot)$ for $D(\cdot, \cdot)$, which are as follows:

a) Dynamic Time Warping (DTW) Distance: The DTW distance, introduced by [Sakoe and Chiba \(1978\)](#), measures similarity by allowing flexible nonlinear alignment between two time series along the temporal dimension. DTW effectively captures similar patterns occurring at different speeds or times by stretching or compressing segments of the time series to find an optimal alignment.

This feature is particularly beneficial in financial markets, where stocks may

exhibit similar trends asynchronously.

DTW uses dynamic programming to construct a cumulative distance matrix and identify the optimal alignment path with the minimum total distance between two time series, $x = (x_1, x_2, \dots, x_T)'$ and $y = (y_1, y_2, \dots, y_S)'$. Let us denote the DTW distance by $D_{DTW}(x, y)$, which is defined even in the case of $T \neq S$, but we consider the case of $T = S$. In this paper, x and y correspond to the $T \times 1$ time series data vector of the i th stock price and the $T \times 1$ mean vector in cluster k , respectively.

Let $d_{t,s}$ denote the cumulative distance between x_t and y_s , which is obtained as the followings.

Step 1: Given $t = 1$, we compute $d_{1,s}$ for $s = 1, 2, \dots, S$ as follows:

$$d_{1,s} = \begin{cases} |x_1 - y_1|, & s = 1 \\ |x_1 - y_s| + d_{1,s-1}, & s = 2, 3, \dots, S \end{cases}$$

Step 2: Given t , we compute $d_{t,s}$ for $s = 1, 2, \dots, S$ as follows:

$$d_{t,s} = \begin{cases} |x_t - y_1| + d_{t-1,1}, & s = 1 \\ |x_t - y_s| + \min(d_{t-1,s-1}, d_{t,s-1}, d_{t-1,s}), & s = 2, 3, \dots, S \end{cases}$$

The above procedure is repeated for $t = 2, 3, \dots, T$.

Step 3: The DTW distance between x and y is given by $d_{T,S}$, i.e.,

$$D_{DTW}(x, y) = d_{T,S}.$$

DTW computes the minimal cumulative distance between the two time series by evaluating all possible alignment paths connecting the starting point (x_1, y_1) and the ending point (x_T, y_T) . The paths move to the right, down or diagonal from (x_i, y_i) to (x_T, y_T) .

The flexibility of DTW allows it to capture similarities effectively even in the presence of temporal distortions, which simpler metrics such as Euclidean often fail to accommodate. However, this flexibility comes at the cost of significantly increased computational complexity, particularly for long time series, because the algorithm must evaluate a vast number of potential alignment paths. Moreover, DTW can be sensitive to short-term anomalies, as isolated outliers might disproportionately influence the optimal alignment and lead to an inflated total distance.

b) Shape-Based Distance (SBD): The SBD proposed by Paparrizos and Gravano (2015) measures similarity by comparing the overall shapes of two time series using normalized cross-correlation (NCC). NCC quantifies how closely one series resembles another at various time offsets. Specifically, NCC is defined as:

$$NCC_{xy}(\tau) = \frac{\sum_{t=1}^{T-\tau} (x_t - \bar{x})(y_{t+\tau} - \bar{y})}{\sqrt{\sum_{t=1}^{T-\tau} (x_t - \bar{x})^2 \sum_{t=1}^{T-\tau} (y_{t+\tau} - \bar{y})^2}} \quad (10)$$

for $\tau = 0, \pm 1, \pm 2, \dots$, where τ denotes the time offset. \bar{x} and \bar{y} are the mean values of time series data x and y , respectively. NCC values range from -1 to 1 . The NCC close to 1 indicates a high similarity after optimal alignment, while the NCC near 0 or negative indicates low or inverse similarity. The SBD between time series x and y , denoted by $D_{SBD}(x, y)$, is defined as:

$$D_{SBD}(x, y) = 1 - \max_{\tau} \text{NCC}_{xy}(\tau) \quad (11)$$

Thus, $D_{SBD}(x, y)$ ranges from 0 to 2. Two time series x and y indicate identical shapes after optimal alignment in the case of $D_{SBD}(x, y) = 0$, while they represent completely inverse shapes or trends in the case of $D_{SBD}(x, y) = 2$. By emphasizing overall shape similarity, SBD effectively captures common cyclical patterns and directional trends, making it particularly useful in financial applications where identifying synchronized behaviors is essential. However, this distance metric can be sensitive to short-term fluctuations or noise, potentially reducing its effectiveness in capturing longer-term trend similarities. Therefore, smoothing or filtering techniques are sometimes applied before using SBD to mitigate the influence of short-term volatility and emphasize underlying trends.

Compared to K-means, K-medoids offers enhanced robustness to outliers and compatibility with arbitrary distance metrics, though at the expense of higher computational complexity. Both algorithms require pre-specifying the number of clusters.

In Section 4, (8), (9) + DTW and (9) + SBD are utilized as clustering methods.

3.2. Cluster Evaluation Method: Silhouette Coefficient

Prior to clustering, the clustering quality under different numbers of clusters requires evaluation to determine the optimal number of clusters. To this end, we can use the silhouette coefficient, which provides a widely accepted measure of clustering quality.

The silhouette coefficient is introduced by [Rousseeuw \(1987\)](#), measuring both cohesion within clusters and separation between clusters. For each data point i , the silhouette coefficient $s^{(i)}$ is calculated as:

$$s^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max(a^{(i)}, b^{(i)})} \quad (12)$$

for $i = 1, 2, \dots, n$, where $a^{(i)}$ is the average distance between data point i and all other data points within the same cluster and $b^{(i)}$ is the average distance between data point i and data points in the nearest cluster. When the i th stock is in cluster k , note that the nearest cluster indicates the cluster which includes the cluster center nearest to the i th stock, where cluster k is excluded.

That is, suppose that C_k is defined by a set of the stocks included in cluster k and $n(C_k)$ is given by the number of the stocks in cluster k . $a^{(i)}$ represents the sample average between the i th stock in cluster k and the other stocks in the same cluster k , which is given by:

$$a^{(i)} = \frac{1}{n(C_k) - 1} \sum_{r_j \in C_k \cap r_j \neq r_i} \|r_i - r_j\|_2$$

for $k = 1, 2, \dots, K$ and $i, j = 1, 2, \dots, n$. Note that $s^{(i)} = 0$ when $n(C_k) = 1$, i.e., when the i th stock only is in cluster k .

Let $-k$ be the cluster nearest to the i th stock, C_{-k} be a set of the stocks

included in the cluster nearest to the i th stock, and $n(C_{-k})$ be the number of the stocks included in C_{-k} .

$b^{(i)}$ is given by:

$$b^{(i)} = \frac{1}{n(C_{-k})} \sum_{r_j \in C_{-k}} \|r_i - r_j\|_2$$

for $k = 1, 2, \dots, K$ and $i, j = 1, 2, \dots, n$.

For evaluation of the number of clusters, the average of the silhouette coefficients are used. That is, we choose the K which maximizes the average of $s^{(i)}$ for $i = 1, 2, \dots, n$. The average of the silhouette coefficients ranges from -1 to 1 , where the values close to 1 indicate well-defined and clearly separated clusters and the values close to -1 suggest poor clustering with overlapping clusters.

The main advantage of the silhouette coefficient is its clear interpretability and its balanced consideration of intra-cluster cohesion and inter-cluster separation. However, its effectiveness diminishes in the case of highly overlapping clusters or complex data structures, which are common situations in economic data. It can also be computationally intensive for large datasets.

4. Empirical Studies Using NIKKEI 225 Stock Data

Stock price data of the 225 companies based on the Nikkei stock average are utilized in this paper, which are taken from NEEDS-FinancialQUEST (<https://finquest.nikkeidb.or.jp/ver2/online/>). Using the 225 stock return data from 28 May, 2024 to 17 December, 2024 and the clustering techniques, we perform the portfolio analysis.

The experiment procedure is as follows:

1) All the 225 stock data are divided into two periods. In this paper we call the first 120 trading day period (i.e., 28 May, 2024 to 19 November, 2024) the clustering period, and the last 20 trading day period (i.e., 20 November, 2024 to 17 December, 2024) the prediction period. Note that 20 trading days indicates about one month data. In this paper, we focus on prediction within one month.

2) Using the first 120 daily data, the 225 companies are classified by the data preprocessing technique (1) shown in Section 2 and the clustering algorithm (8) in Section 3. As shown in **Figure 1**, the average silhouette coefficient is drawn for $K = 6, 7, \dots, 12$ to obtain the number of clusters. As a result, $K = 9$ is chosen because the average silhouette coefficient is maximized at $K = 9$.

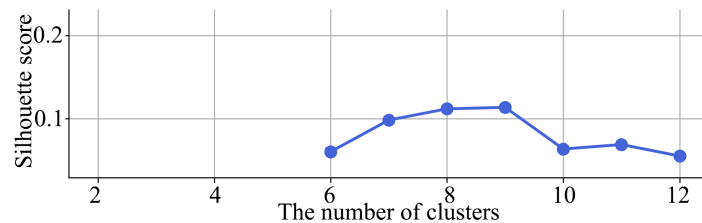


Figure 1. Average silhouette coefficients: The Case of (1) and (8).

3) For the data preprocessing techniques (1) - (7) and the clustering algorithms (8), (9) + DTW and (9) + SBD, the 225 stock return data are classified into $K = 9$ groups using the first 120 trading day data. The average returns and the average standard deviations are computed for both the clustering and prediction periods of each group.

4) We examine whether we can forecast the prediction period from the clustering period. If we can forecast the stock returns and their standard deviations (i.e. volatilities), we might observe that there are strong positive correlations between the clustering period and the predicted period regarding the average return and the average standard deviation. It is well known that stock returns are skewed to the left (i.e., skewness is negative) in most cases and are fatter than normal distribution in tails (i.e., kurtosis is greater than 3). That is, stock returns are not normally distributed. Accordingly, without specifying any distribution, we utilize the rank correlation to examine whether there is a correlation between the two periods. Thus, for both periods, each cluster is ranked based on the average returns and the average standard deviations.

As mentioned above, it is well known that stock return data are not normally distributed, i.e., negative skewness and large kurtosis with comparison of normal distribution. In addition, critical values are explicitly obtained in the case of rank correlation. Therefore, it is more relevant to utilize the ranked data rather than the original return data. Moreover, if rank correlation is large, we can choose high-ranked stocks in the next period, which is of advantage for portfolio.

In general, nonparametric tests have less power than parametric ones. However, it is well known that asymptotic efficiency of nonparametric tests does not show so bad performance (for example, see [Hodges and Lehman \(1956\)](#) and [Chernoff and Savage \(1958\)](#)). In addition, according to [Tanizaki \(1997\)](#), the Wilcoxon test is more powerful than the t -test even in the case of small sample and non-Gaussian sample. Therefore, in this paper we utilize the rank correlation and also the $K = 9$ clusters might be enough.

AVE in [Table 1](#), denotes the rank correlation of the average stock returns in each cluster between two periods (i.e., clustering period and prediction period). If the high return cluster during the clustering period still stays in the high return cluster during the prediction periods, it might be better to purchase the stocks included in the high return cluster during the clustering period. That is, in this case we can forecast the stock prices (i.e., the market efficiency hypothesis in stock market is rejected). However, if there is no rank correlation between the average stock returns in each cluster between the two periods, we can conclude that we cannot forecast the future stock prices given the past ones (i.e., the market efficiency hypothesis is supported). As a result, the rank correlations of AVE are insignificant in all the cases of (1) - (7) for (8), (9) + DTW and (9) + SBD. That is, we can conclude that the stock market in Japan is efficient at least during the period from 28 May, 2024 to 17 December, 2024.

Table 1. Rank correlations between clustering and prediction periods.

Data Pre-Processing Techniques	Clustering Methods					
	(8)		(9) + DTW		(9) + SBD	
	AVE	SER	AVE	SER	AVE	SER
(1)	-0.133	0.433	0.050	-0.283	0.450	0.667**
(2)	-0.133	0.550**	0.067	0.150	-0.200	0.500*
(3)	0.333	0.250	0.200	0.250	-0.183	0.567*
(4)	0.000	0.483*	-0.083	0.450	0.333	0.283
(5)	0.283	0.517*	-0.517	-0.083	0.217	0.350
(6)	-0.367	0.117	0.200	0.317	-0.333	0.600**
(7)	0.117	0.400	0.017	0.067	-0.050	0.683**

The clustering period is 2024/05/28-2024/11/19 (120 trading days) and the prediction period is 2024/11/20-2024/12/17 (20 trading days). In the case of the rank correlation with K = 9, top 1%, 5% and 10% points are given by 0.7667, 0.5833 and 0.4667, which are denoted by ***, ** and * in the above table, respectively.

In **Table 1**, SER indicates the rank correlation of the average standard deviations (or volatilities) in each cluster between two periods. In the case of financial data, the generalized auto-regressive conditional heteroscedasticity model (hereafter, GARCH) and the stochastic volatility model (hereafter, SV) are often utilized. These models represent that variances are time-dependent (especially, the present variance depends on the past one). That is, this fact indicates that we can forecast the variance (or standard deviation) given past one. It is known that variance is a measure of risk, i.e., a measure of market instability. Therefore, it might be important to know the movements of variances (i.e., volatilities). In this paper, we might obtain the results that volatilities are persistent because the rank correlations of SER are significantly greater than zero in some cases (at least, the rank correlations of SER are larger than those of AVE in a lot of cases).

Next, in **Table 2** we take the three estimation periods and 100 trading days in addition to 120 trading days for the clustering period (or training period), in which (8) is taken for the clustering method. The column of 24/05/28-24/11/19 (120) and 24/11/20-24/12/17 (20) is exactly same as (8) in **Table 1**, which is put for comparison with the other two columns. The results are almost same as **Table 1**. A lot of SERs are significantly greater than zero while AVRs are not positive at all. That is, the Nikkei 225 stock market is efficient and accordingly the stock prices are not predictable. In contrast, stock price volatilities depend on the lagged ones, which implies that GARCH or SV models are appropriate for estimation of stock returns.

Table 2. Rank correlations between clustering and prediction periods.

		Clustering Method (8)					
Data	24/05/28-24/11/19 (120)	24/11/28-25/05/29 (120)		25/04/01-25/08/25 (100)			
	24/11/20-24/12/17 (20)	25/05/30-25/06/26 (20)	25/08/26-25/09/24 (20)				
Pre-Processing							
Techniques	AVE	SER	AVE	SER	AVE	SER	
(1)	-0.133	0.433	0.183	0.150	0.450	0.933***	
(2)	-0.133	0.550**	-0.267	0.267	-0.700	0.483*	
(3)	0.333	0.250	-0.083	0.200	-0.467	0.467*	
(4)	0.000	0.483*	0.000	0.317	0.300	0.600**	
(5)	0.283	0.517*	0.067	0.583**	0.250	0.767***	
(6)	-0.367	0.117	0.233	0.167	0.350	0.250	
(7)	0.117	0.400	-0.433	0.067	0.433	0.750**	

5. Summary and Concluding Remarks

To examine whether the financial market is efficient, we usually estimate the autoregressive process with other explanatory variables, taking into account time-varying heteroscedastic error terms such as GARCH or SV.

In this paper, we take the different approaches where the cluster techniques are utilized. We construct K clusters ($K = 9$ is taken in this paper), rank the average returns of the K clusters in order of size, and similarly rank them during prediction period. We check whether the ranks in the clustering period correspond to those in the prediction period. Using the rank correlation, we test the correspondence between the ranks in the clustering and prediction periods. If the ranks in the clustering period are similar to those in the prediction period, we can conclude that we can forecast the stock returns given past information. As a result, it might be concluded from AVE in **Table 1** that we cannot predict future stock returns given past stock returns, i.e., clustering with the present data do not depend on clustering with the past data, because all the rank correlations of AVE are statistically zero. This result is consistent with the market efficiency hypothesis.

In addition, the same procedure is performed regarding the standard deviations (i.e., volatilities). That is, we rank the average standard deviations of the K clusters in order of size, similarly rank them during prediction period, and compare both ranks. In the case of financial data, GARCH and SV are frequently utilized, which describe that the present variance explicitly depends on the past one. In other words, it is well known that the present volatility influences the future volatility. In this paper, we obtain similar results through cluster analysis.

In **Table 2**, taking different sample periods, we investigate whether stock returns and their volatilities are predictable or not by clustering. The results are very similar to **Table 1**.

Thus, in this paper, we can conclude that stock returns volatilities are predictable but stock returns are not. These results are consistent with the past studies.

Acknowledgements

This research is partially supported from Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (C) 22K01423.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- Brida, J. G., & Risso, W. A. (2009). Dynamics and Structure of the 30 Largest North American Companies: A Minimal Spanning Tree Study. *Computational Economics*, *35*, 85-99. <https://doi.org/10.1007/s10614-009-9187-1>
- Chen, Y., Xu, R., Wang, J., Yang, H., & Wang, X. (2021). Clustering Financial Time Series to Generate a New Method of Factor Neutralization: An Empirical Study. *International Journal of Financial Engineering*, *8*, Article 2141005. <https://doi.org/10.1142/s242478632141005x>
- Chernoff, H., & Savage, I. R. (1958). Asymptotic Normality and Efficiency of Certain Non-parametric Test Statistics. *The Annals of Mathematical Statistics*, *29*, 972-994. <https://doi.org/10.1214/aoms/1177706436>
- Chun, D., Cho, H., & Ryu, D. (2025). Volatility Forecasting and Volatility-Timing Strategies: A Machine Learning Approach. *Research in International Business and Finance*, *75*, Article 102723. <https://doi.org/10.1016/j.ribaf.2024.102723>
- Drago, C. (2024). *Ensemble Financial Time-Series Clustering*. https://www.researchgate.net/publication/379084476_Ensemble_Clustering_of_Financial_Time_Series
- Hodges, J. L., & Lehman, E. L. (1956). The Efficiency of Some Nonparametric Competitors of the t-Test. *The Annals of Mathematical Statistics*, *27*, 324-335. <https://doi.org/10.1214/aoms/1177728261>
- Kaufman, L., & Rousseeuw, P. J. (1987). Clustering by Means of Medoids. In Y. Dodge (Ed.), *Statistical Data Analysis Based on the L1-Norm and Related Methods* (pp. 405-416). Springer.
- Liao, T. W. (2005). Clustering of Time Series Data—A Survey. *Pattern Recognition*, *38*, 1857-1874. <https://doi.org/10.1016/j.patcog.2005.01.025>
- López-Oriona, Á., Montero-Manso, P., & Vilar, J. A. (2025). Time Series Clustering Based on Prediction Accuracy of Global Forecasting Models. *Knowledge-Based Systems*, *323*, Article 113649. <https://doi.org/10.1016/j.knosys.2025.113649>
- MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, *5*, 281-297.
- Mantegna, R. N. (1999). Hierarchical Structure in Financial Markets. *The European Physical Journal B*, *11*, 193-197. <https://doi.org/10.1007/s100510050929>
- Marti, G., Andler, S., Nielsen, F., & Donnat, P. (2016). Clustering Financial Time-Series: How Long Is Enough? In *Proceedings of the 25th International Joint Conference on Artificial Intelligence* (pp. 2583-2589). AAAI Press.
- Paparrizos, J., & Gravano, L. (2015). K-Shape: Efficient and Accurate Clustering of Time-Series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 1855-1870). ACM. <https://doi.org/10.1145/2723372.2737793>

- Paparrizos, J., Morvant, E., & Theodoridis, S. (2024). *Bridging the Gap: A Decade Review of Time-Series Clustering Methods*. <https://arxiv.org/abs/2412.20582>
- Rousseeuw, P. J. (1987). Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, 20, 53-65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)
- Sakoe, H., & Chiba, S. (1978). Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26, 43-49. <https://doi.org/10.1109/tassp.1978.1163055>
- Shi, B., & Xiao, M. (2022). *Time-Series K-Means in Causal Inference and Mechanism Clustering for Financial Data*. <https://arxiv.org/abs/2202.03146>
- Tan, Y., Tan, Z., Tang, Y., & Zhang, Z. (2024). Functional Volatility Forecasting. *Journal of Forecasting*, 43, 3009-3034. <https://doi.org/10.1002/for.3170>
- Tanizaki, H. (1997). Power Comparison of Non-Parametric Tests: Small-Sample Properties from Monte Carlo Experiments. *Journal of Applied Statistics*, 24, 603-632. <https://doi.org/10.1080/02664769723576>
- Tola, V., Lillo, F., Gallegati, M., & Mantegna, R. N. (2008). Cluster Analysis for Portfolio Optimization. *Journal of Economic Dynamics and Control*, 32, 235-258. <https://doi.org/10.1016/j.jedc.2007.01.034>
- Tumminello, M., Lillo, F., & Mantegna, R. N. (2010). Correlation, Hierarchies, and Networks in Financial Markets. *Journal of Economic Behavior & Organization*, 75, 40-58. <https://doi.org/10.1016/j.jebo.2010.01.004>
- Zhang, C., Zhang, Y., Cucuringu, M., & Qian, Z. (2024). Volatility Forecasting with Machine Learning and Intraday Commonality. *Journal of Financial Econometrics*, 22, 492-530. <https://doi.org/10.1093/jjfinec/nbad005>