國立中山大學資訊管理學系研究所

碩士論文

Department of Information Management

National Sun Yat-sen University

Master Thesis

以獨立成分分析為基之支援向量迴歸模式預測時間系列股價

ICA-clustered Support Vector Regressions in Time Series Stock Price Forecasting

研究生：陳則丞

Tse-Cheng,Chen

指導教授：張德民 博士

Dr. Te-Min Chang

中華民國 101 年7 月

July 2012

國立中山大學研究生學位論文審定書

本校資訊管理學系碩士班

研究生陳則丞（學號：M994020040）所提論文

以獨立成分分析為基之支援向量迴歸模式預測時間系列股價
ICA-clustered Support Vector Regressions in Time Series Stock Price Forecasting

於中華民國 101 年 7 月 31 日經本委員會審查並舉行口試，符合碩士學位論文標準。

學位考試委員簽章：

召集人 黃三益 黃三益　　委　員 張德民 張德民

委　員 蕭文峰 蕭文峰　　委　員 _____

委　員 _____　　委　員 _____

指導教授(張德民) 張德民 (簽名)

誌　謝

有機會於中山資管就讀碩士班，是一件很幸福的事，兩年的歷程中有汗水、有歡笑，這一切的過程終於在兩年後的今天有了豐碩的果實；而能夠有這樣的成果，實是歸功於許多人的傾囊相助。最首先感謝的是我的指導教授張德民老師，在這兩年期間對學生的諄諄教誨，學生知道自己在學習研究上有許多方面都有所不足，但您還是願意有教無類的對學生進行指導，讓學生能夠在兩年後的今天完成碩士畢業論文；老師在學習的過程中每個細節的指點，都是讓學生很難忘的記憶，老師辛苦您了！接著要感謝的是我的父母，感謝我的父母在養育我二十餘年後，還願意讓我再多花兩年的時間於學習上面；並且無條件、無怨言的支持我完成學業，烏鴉尚知反哺，何況身為人? 父母如此的恩情，孩兒在有生之年必當全力回報。

接著要感謝的是實驗室的同學立人與學弟權益，有立人在這兩年來的一同進步與課業、研究上的討論，以及權益於實驗室大小事務上的幫忙，才使我能夠順利的完成學業，謝謝你們！再來絕對不會忘的是於 1029 Lab 一起過了兩年的夥伴們，我們一起奮鬥、一起玩樂，一起面對在碩士班中的所有忙碌；包含課業與每一個所上的活動，感謝常處於同一間 Lab 的德媽、盧小小、小賴、花生、小峰、罐子、TORO、拉拉、許咪、金毛、BBB，以及常常將 1029 當作交誼廳來談正事或聊聊天的 JJ、大門、大頭、鳳梨、SOSO、菻菻、OTT、俊傑、科翰、大豆、小馬，因為有你們，我才能夠在生活中還能夠有這麼多的樂趣與難忘的回憶，能認識你們真好，以及每一位文中沒提到，但我絕不會忘記的同學們。最後要感謝的是一同到中山就讀碩士班的小哈，因為你，我才深刻的感受到人生的態度該是什麼，到了碩士班後我們相處的時間不多，但我會永遠記得你對於任何事物的熱忱與積極，並以此為座右銘而努力！

天下碩士班如此之多，何其幸運能夠就讀中山資管，使我這兩年的收穫不只是研究成果與一張文憑，更有許多回憶以及對於未來的啟發，謹以此篇誌謝詞紀念兩年來的一切。

則丞

於中山大學

中華民國一零一年七月

摘　要

財務時間序列資料預測長久以來一直被廣泛地研究探討，因為它是投資者做投資決定的重要依據。然而，這樣的工作有極大的挑戰，因為財務時間序列資料是混雜且不穩固的。過去這方面的研究以類神經網路為主，但是這類方法有著模式解釋力與預測一般化能力不足的問題。

因此本研究的目的在於提出一個混合式方法預測公司股價。本研究提出使用獨立成分分析來瞭解時間序列資料的潛在結構，以及去除資料雜訊，並且以其結果進行分群分析；接著使用支援向量迴歸對每一群資料建立一個預測模式，以冀提升其一般化能力。

本研究進行兩個實驗來驗證所提方法，實驗結果顯示支援向量迴歸有穩健的績效表現。而更重要的是，利用獨立成分分析分群資料建立的支援向量迴歸模式確實比不分群資料建立的支援向量迴歸模式表現較佳。這顯示了我們所提方法可以提升預測模式一般化的能力，也因此驗證了我們所提方法在實際應用上的可行性。

關鍵字：　財務時間序列資料、時間序列資料預測、獨立成分分析、分群分析、支援向量迴歸

Abstract

Financial time-series forecasting has long been discussed because of its vitality for making informed investment decisions. This kind of problem, however, is intrinsically challenging due to the data dynamics in nature. Most of the research works in the past focus on artificial neural network (ANN)-based approaches. It has been pointed out that such approaches suffer from explanatory power and generalized prediction ability though.

The objective of this research is thus to propose a hybrid approach for stock price forecasting. Independent component analysis (ICA) is employed to reveal the latent structure of the observed time-series and remove noise and redundancy in the structure. It further assists clustering analysis. Support vector regression (SVR) models are then applied to enhance the generalization ability with separate models built based on the time-series data of companies in each individual cluster.

Two experiments are conducted accordingly. The results show that SVR has robust accuracy performance. More importantly, SVR models with ICA-based clustered data perform better than the single SVR model with all data involved. Our proposed approach does enhance the generalization ability of the forecasting models, which justifies the feasibility of its applications.

Keywords: financial time-series, time-series forecasting, independent component analysis, cluster analysis, support vector regression

Table of contents

List of figures

List of tables

Chapter 1 Introduction

1.1 Overview

Financial time-series forecasting has long been regarded as a challenging research work. It aims at analyzing the historical time-series data to predict the future time-series results such as stock prices and stock index. This kind of problem is intrinsically difficult because the time-series data are of noisy, non-stationary, and chaotic characteristics, which drives the forecasting models prone to errors.

In the past, artificial neural networks (ANNs) have been applied in time-series forecasting to capture functional relationships between the empirical input-output data. ANNs are universal function approximators that are capable of mapping any nonlinear function without assumptions of the data characteristics. However, the internal operations of ANNs act like a black box where the underlying relationships of the input-output data are unknown or hard to describe. Moreover, the trained ANN models lead themselves easily to over-fitting due to the presence of data noise. Therefore, techniques to uncover the latent structure of the financial time-series data, and enhance the generalization ability of the forecasting models are essential to develop beyond ANNs.

In recent years, a new technique, known as *independent component analysis* (ICA) has been proposed to solve the so-called blind source separation problem (Hyvärinen

and Oja, 2000). It is a statistical signal processing technique to extract independent sources from observed data that are mixtures of the unknown sources without knowing the mixing mechanisms. Namely, the goal of ICA is to separate the observed mixing signal into unobserved original sources by minimizing their statistical dependence. Those unobserved sources are called independent components (ICs).

ICA has been widely applied in such fields as medical signal processing (e.g. electroencephalography analysis), audio signal processing (e.g. music information retrieval), pattern recognition (e.g. face recognition), and so on. Not surprisingly, it can also be applied to financial time-series data analysis since an observed multivariate time series (such as daily stock price) reveal the reaction (mixture) of a system (such as the stock market) to a few statistically independent time series. By exploring the relationships between the observed time series and the unobserved independent time-series components, the latent structure of the observed data can be uncovered and further be cleanup by removing unessential components. With the less noisy latent structure, the forecasting task can be facilitated to enhance its prediction performance.

On the other hand, support vector machine (SVM) based on statistical learning theory is proposed for classification tasks (Vapnik, 1995). Later on, with the introduction of ε–intensive loss function[1], SVM evolves into the non-linear regression

[1] This is the most well-known type of loss function, but not the only one.

model, called support vector regression (SVR), for prediction tasks. Both SVM and SVR have received increasing attention from researchers because the *Structural Risk Minimization Principle* is adopted to minimize the upper bound of the generalization error, whereas most traditional machine learning techniques (including ANNs) adopt the *Empirical Risk Minimization Principle* to minimize the upper bound of the training error. Therefore, SVM and SVR possess better generalization ability than traditional techniques, and particularly, SVR is expected to perform better in the financial time-series forecasting problem than ANNs.

1.2 Objective of the research

Based upon the above descriptions, the objective of this research is thus to propose a hybrid stock price forecasting approach that makes use of ICA and SVR. ICA analysis is first applied to reveal the latent structure of observed stock prices and help in cleaning-up the structure. With the processed latent structure, companies under study can be grouped into clusters where companies in a cluster exhibit a similar stock price trend and companies in distinct clusters have different stock price patterns. We then build separate SVR models for companies within each individual cluster. The reason behind this lies in that stock price trends of distinct companies vary considerably and a single SVR model built based on all companies may not be reliable and robust. In contrast, the SVR model

tends to be more robust if it is built based on companies that exhibit a similar price trend. Consequently, the overall performance from all SVR models built this way is expected to be better in the forecasting task.

1.3 Organization of the research

This chapter introduces the overview and the objective of our research study. The rest of the thesis is organized as follows. Chapter 2 reviews literature related to this research, which includes ICA analysis, clustering techniques, and support vector regression. Our proposed hybrid approach is discussed in details in chapter 3. Chapter 4 illustrates the experiments and corresponding results to examine the performance of our proposed approach. Finally, concluding remarks and future work are addressed in Chapter 5.

Chapter 2 Literature Review

Relevant literature to our research is reviewed in this chapter. Section 2.1 introduces independent component analysis that extracts statistically independent components from time-series data, followed by the discussion of clustering analysis. Finally, support vector regression for prediction tasks is illustrated in Section 2.3.

2.1 Independent Component Analysis

In real world applications, one often faces the situation of big data that are of multivariate. Directly manipulating the multivariate data is a challenging task. Therefore, finding a suitable representation of those data is desired for computational and conceptual simplicity purpose. The representation is commonly sought as a linear transformation of the original data such as principal component analysis (PCA).

PCA adopts linear algebra techniques to convert observations of possibly correlated variables into uncorrelated ones, called *principle components*. The first principle component found accounts for as much of the variability in the data as possible, and the second principle component is orthogonal to the first one and captures as much of the remaining variability as possible, and so on. The goal of PCA is to find a new set of variables that capture as much of the variability as possible while discarding less important ones as many as possible. Therefore, the number of principle components is

generally much less than the number of original variables in real applications.

Even though PCA serves as a multivariate representation technique with dimension reduction, its methodological foundation from linear algebra is unsound in the statistical view. On the other hand, independent component analysis (ICA) is developed by Hyvärinen and Oja (2000) to find a linear representation of non-Gaussian data so that the extracted components are statistically independent.

ICA was originally proposed to deal with the blind source separation (BSS) problem that deduces an observed mixing signal into several unobserved original signals. The "blind" in BSS means that we know little about the source of the mixing signals, which means that either we do not know the original signals in the mixing matrix, or how these signals are mixed into the mixing matrix (Cardoso, 1998).

To deal with BSS problem, ICA separates the observed mixing signal into statistically components which are called *independent components* (ICs). These components can be used to estimate the original signals even though they are not the true ones. Therefore, ICA aims at retrieving unknown independent components (ICs) that are equivalent to original signals from an observed mixing signal Naik and Kumar (2011).

Ever since the introduction of ICA, a variety of applications have been resulted. It has been applied to such fields as content-based speech recognition (Chien and Chen, 2006), Electroencephalogram analysis (Lin, 2005; Lin, 2006), image processing (Le

Borgne and Guérin-Dugué, 2004), and time-series financial data applications (Back and Weigend, 1997).

Back and Weigend (1997) discussed the application of ICA to multivariate financial time series data such as a portfolio of stocks. Their idea was to factorize the portfolio and extract independent components. Data of the largest 28 Japanese stocks were collected and the analysis showed that the estimated ICs fell into two categories as infrequent but large shocks, and frequent smaller fluctuations. In addition, they showed that the overall stock price could be reconstructed well by using a small number of weighted ICs. Therefore, it was concluded that ICA was a potentially powerful method of analyzing and understanding financial data.

In the following subsections, we will briefly introduce the ICA model, its preprocessing procedure, the FastICA algorithm proposed for extracting the ICs, and methods to select important ICs.

2.1.1 Definition of ICA

To define ICA precisely, we can adopt a statistical generative model where latent variables are involved. Let $\boldsymbol{x(t)}$ be the set of observed mixing signals and $\boldsymbol{s(t)}$ be the set of independent components, also called signal sources. Assume that we observe n linear mixtures $x_1(t),..., x_n(t)$ of n independent components $s_1(t),\ldots, s_n(t)$. Their relationship can

then be expressed as

$$\boldsymbol{x}(t) = \mathbf{A}\,\boldsymbol{s}(t) \tag{2-1}$$

where matrix **A** represents a *mixing matrix*, and both **A** and $\boldsymbol{s}(t)$ are unknown.

The basic assumptions behind $\boldsymbol{s}(t)$ is that the components are statistically independent. Furthermore, the independent components must have *non-Gaussian* distributions because the matrix **A** is not identifiable for Gaussian independent components. To solve $\boldsymbol{s}(t)$, we consider a linear combination of the x_i's, and denote this by

$$\boldsymbol{y}(t) = \mathbf{W}^T \boldsymbol{x}(t) \tag{2-2}$$

where **W** is a vector to be estimated. If **W** were one of the rows of the inverse of **A**, this linear combination would actually equal one of the independent components, i.e, $y(t)$ would be equivalent to one of the $s_i(t)$ for some i. This can be achieved by maximizing the non-Gaussianity of $\mathbf{W}^T \boldsymbol{x}(t)$ so that $y(t)$ equals to one of the independent components. Therefore, the issue to find ICs is to estimate the matrix **W**, also called the de-mixing matrix, that maximizes the non-Gaussianity of $\mathbf{W}^T \boldsymbol{x}(t)$.

According to the above descriptions, the ICA procedure can be illustrated in Figure 2-1, where $\hat{s}(t)$ stands for the ICs obtained by ICA.

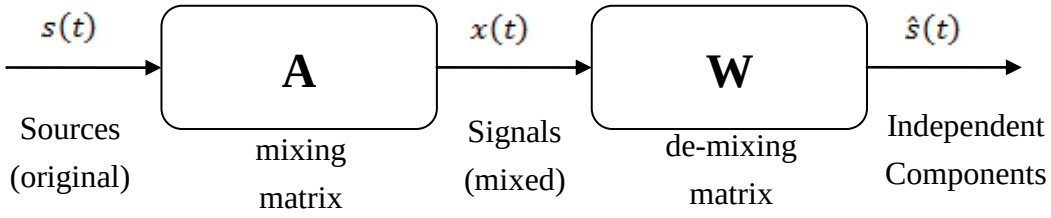


Figure 2- 2 Basic ICA procedure (Back & Weigend, 1997)

2.1.2 Preprocessing of ICA

Before applying the ICA algorithm on the data, it is useful to perform some preprocessing. These preprocessing techniques make the problem of ICA estimation simpler and better conditioned (Hyvärinen and Oja, 2000), which includes *centering* and *whitening*.

Centering

This preprocessing is done solely to simplify the ICA algorithms. To center ***x*** is to subtract its mean, E(Cheung & Xu) from ***x*** so that it becomes "zero-mean" as

$$\boldsymbol{x}' = x - E(\text{Cheung \& Xu}) \tag{2-3}$$

Note that by doing so, ***s*** will be zero-mean as well. After estimating the de-mixing matrix **W**, we can complete the estimation by adding the mean vector of ***s*** back to the centered estimates of ***s***.

Whitening

Whitening is to transform the observed vector $\boldsymbol{x}$ linearly to obtain a new vector $\tilde{\mathbf{x}}$ which is white, i.e. the components are uncorrelated and their variances equal unity. In other words, the covariance matrix of $\tilde{\mathbf{x}}$ equals the identify matrix as

$$\mathrm{Cov}(\tilde{\mathbf{x}}) = \mathrm{E}\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathrm{T}}\} = \mathbf{I} \tag{2-4}$$

2.1.3 The FastICA Algorithm

Hyvärinen and Oja (1997) has proposed the FastICA algorithm to solve $\mathbf{W}$ based on a fixed-point iteration scheme to find a maximum of the non-Gaussianity of $\mathbf{W}^T \boldsymbol{x}$. Especially, negentropy that is based on the information-theoretic quantity of differential entropy is employed as a good measure of non-Gaussianity. This quantity can be approximated by

$$[E\{G(y)\}-E\{G(\nu)\}]^2 \tag{2-5}$$

where G is any nonquadratic function and ν is a Gaussian variable of zero mean and unit variance (i.e., standardized). Denoting g as the derivative of the nonquadratic function G, the FastICA algorithm for one $x(t)$ is as follows:

1. Choose an initial (e.g. random) weight vector $\mathbf{w}$.

2. Let $\mathbf{w}^+ = E\{\boldsymbol{x}g(\mathbf{w}^T \boldsymbol{x})\}-E\{g'(\mathbf{w}^T \boldsymbol{x})\}\mathbf{w}$

3. Let $\mathbf{w} = \mathbf{w}^+ / \left\|w^+\right\|$

4. If not converged, go back to 2.

In practice, the expectations in FastICA will be replaced by their estimates, which are naturally of course the corresponding sample means.

2.1.4 IC Selection

As stated, the maximum number of independent components is equal to the number of observed signals. Like PCA, some of them are less important to contribute to the latent structure of the observed signals and can be ignored. The number of essential principle components is generally much less than the maximum number. However, unlike PCA that importance of principal components can be indicated from the corresponding eigenvalues to reflect the explained variability, currently there does not exist a systematic way for ICA to choose essential independent components.

The most simple and straightforward method to select a subset of ICs is to enumerate all kinds of combinations of ICs to examine how the combinations are close to the original observed signals under some sort of distance measures. However, the exhaustive search is costly in computational efforts. Cheung and Xu (2000) (Cheung & Xu, 2001)proposed a sub-optimal search approach to ordering the ICs in a greedy Testing-and-Acceptance (TnA) manner. A relative Hamming distance (RHD) was defined to measure the difference between the combinations of selected ICs and the

original signals. Then the ICs were ordered in a reverse way that each time the least important IC was taken out from the current combination consideration. The resultant list of ordered ICs indicated the ICs' importance in descending order. This approach, though sub-optimal in the nature, is much more efficiency than the exhaustive search.

2.2 Cluster Analysis

Cluster analysis is a technique that groups data objects together based on their relationships. Usually, it is desired for the clustering results that data objects within a cluster should be similar to one another as much as possible, while the individual clusters should be distinct enough (Tan et al., 2006).

Clustering techniques can be usually classified into hierarchical and partitional ones. Hierarchical clustering seeks to build a hierarchy of clusters. The strategy for hierarchical clustering generally falls into two types: agglomerative and divisive.

Agglomerative hierarchical clustering adopts the strategy of a bottom-up approach where each data point is deemed as one cluster initially. The clustering algorithm will merge these clusters together according to their similarity, and iterate itself until there is one cluster containing all data in the last iteration. In contrast, the divisive hierarchical clustering adopts the strategy of a top-down approach where all data points are included in one cluster initially. The clustering algorithm will separate the cluster into several

clusters, until each cluster contain one single data point.

In hierarchical clustering techniques, we need to further define the promixity measure between two clusters. Common promixity measures include the single-link, complete-link, and Ward's method. The single-link measure defines promixity of two clusters as the minimum distance between their respective data points, while the complete-link measure defines promixity as the maximum distance between their respective data points.

Ward's method is a measure between the minimum distance and the maximum distance. The distance between two clusters A and B is the increase of the sum of squared errors (SSE) due to the merge of the two cluster, which can be defined as follows:

$$\Delta(\mathrm{A},\ \mathrm{B}) = \frac{n_A n_B}{n_A + n_B} \left\| m_A - m_B \right\|^2 \qquad (2\text{-}6)$$

Where n_A, n_B represent the number of data points in clusters A and B, respectively, m_A, m_B represent the centroids of clusters A and B, respectively, and Δ is the merging cost of combining the two clusters.

The result of hierarchical clustering can be represented by a dendrogram or a nested partition graph, which can help us to observe the structure of clustering hierarchy. Generally speaking hierarchical clustering has its own advantages and disadvantages. On the bright side, it avoids the difficulty of clustering optimization problem, and does

not need to choose initial data points as clusters. It, however, takes a prohibitive time complexity to operate.

The second type of clustering techniques is the partitional method, which attempts to directly separate the data set into a set of disjoint clusters. K-means method is one of the most popular techniques of this type (MacQueen, 1967). Given a set of raw observations (x_1, x_2, …,x_n) where each observation is a *d*-dimensional vector, K-means method starts the following procedure with the number of clusters known:

1. Begin with an initial partition of the data set into the specified number of clusters; compute the centroids of these clusters.
2. Assign each data point to the cluster whose centroid is the nearest from this data point.
3. Compute the new centroids of the clusters after all data are reallocated.
4. Repeat steps 2 and 3 until no data points change the cluster assignments.

In fact, a more difficult question to answer in K-means is how to determine the number of clusters. Most of the research works follow a trial and error method to test several possible numbers of clusters. If the numbers of clusters are specified, the measure of the *sum of the squared error* (SSE) can be applied to determine which clustering result is better. The SSE is formally defined as follows:

$$\text{SSE} = \sum_{i=1}^{\text{K}} \sum_{x \in C_i} dist(C_i, x)^2 \qquad (2\text{-}7)$$

where the distance measure *dist* is usually of the Euclidean metric.

2.3 Support Vector Regression

Classification is an important research issue in data mining or text mining fields. Many techniques have been developed to perform classification tasks. Among others, the most significant technique is support vector machine (SVM) that has exhibited its power of classification accuracy over many classifiers (Vapnik, 1995).

The primitive support vector machine is a linear classification method. A hyperplane to separate the binary class data is defined as

$$<w, x> + b = 0 \tag{2-8}$$

where w is the normal (coefficient) vector to x, b is the y-intercept, and $<\bullet>$ denotes the inner product of two vectors. SVM aims at finding the hyperplane to separate the nearest data class with the largest margin, called the optimal separating hyperplane (OSH). The two hyperplanes located on the largest margin and paralleled with OSH, are called support hyperplanes. The solutions of w and b in this case are expressed as

$$w^* = \Sigma\, \alpha_i\, y_i\, x_i$$

$$b^* = -\frac{1}{2}\ <w^*, x_r + x_s> \tag{2-9}$$

where α are the Language multipliers, and x_r and x_s are support vectors from each class.

In general cases, the training data may not be completely linearly separable.

Therefore, an additional cost function is associated with the misclassifications where the cost C is assigned. The solutions of *w* and *b* in this case are the same as in (2-9) with α_i subject to the following constraint:

$$0 \leq \alpha_i \leq C \tag{2-10}$$

However, in practice, most of the classification tasks are not linearly separable. SVM introduces kernel functions that act as if the input data were transformed into a higher dimensional feature space and a linear separation was performed in that higher dimensional space. The solutions of *w* and *b* in this case with cost function considered are expressed as

$$<w^*, x> = \Sigma\, \alpha_i\, y_i\, K(x_i, x)$$

$$b^* = -\frac{1}{2} \Sigma\, \alpha_i\, y_i\, [K(x_i, x_r) + K(x_i, x_s)]$$

with constraints $0 \leq \alpha_i \leq C$ (2-11)

where $K(\bullet)$ is the kernel function.

On the other hand, support vector regression (SVR), one of the variants of SVM, is developed to deal with regression (or prediction) problems (Vapnik et al., 1997). The basic formulation of SVR is similar to that of SVM. Nevertheless, in SVR, a loss function is introduced to tolerate errors in the predicted results. The most common loss function is the ε-intensive loss function, which is illustrated in Figure 2-2.

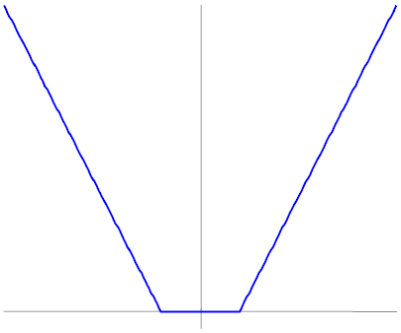


Figure 2-2 ε-intensive loss function

The solutions of *w* and *b* in the linear regression case with the cost function considered are expressed as

$$w^* = \Sigma\,(\alpha_i - \alpha_i^*)\,x_i$$

$$b^* = -\frac{1}{2}\langle w^*, (x_r + x_s)\rangle$$

with constraints $0 \le \alpha_i, \alpha_i^* \le C$ (2-12)

where α_i and α_i^* are Language multipliers such that $\Sigma\,(\alpha_i - \alpha_i^*) = 0$.

Finally, the solutions of *w* and *b* in the non-linear regression case are expressed as

$$\langle w^*, x\rangle = \Sigma\,(\alpha_i - \alpha_i^*)\,K(x_i, x)$$

$$b^* = -\frac{1}{2}\Sigma\,(\alpha_i - \alpha_i^*)\,[K(x_i, x_r) + K(x_i, x_s)]$$

with constraints $0 \le \alpha_i, \alpha_i^* \le C$ (2-13)

The most common kernel function employed is the exponential radial basis function (ERBF), which is defined as

$$K(x, x') = \exp\left(-\frac{\|x - x'\|}{2\gamma^2}\right) \quad (2\text{-}14)$$

where γ denotes the radius that has to be estimated.

In summary, before we solve *w* and *b* in a typical SVR formulation, three parameters need to be assigned, i.e. ε, C, and γ. These parameters vary dramatically from domain to domain. For example, Lu and Wang (2010) proposed an approach which concerned the product demand forecasting. In this work, the parameters were set to be $\varepsilon = 2^{-9}$, $C = 2^{3}$, and $\gamma = 2^{-1}$. On the other hand, Cao and Tay (2001) studied financial forecasting and chose the parameters to be $\varepsilon = 2^{-3}$, $C = 10$, and $\gamma = 0.015$. Based on these observations, somehow, the determination of those parameters may itself become a research issue in SVR applications.

Chapter 3 Proposed Approach

The objective of this research is to propose a hybrid stock price forecasting model that makes use of ICA for clustering and applies SVR for prediction. The basic idea behind it is that stock price trends of different firms vary significant and thus it makes no sense to predict them as a whole with a single model. In contrast, firms with similar stock price trends can be grouped as clusters. It makes more sense to predict the stock prices for firms in the same cluster with a single model, and develop different models for different clusters.

Consequently, we adopt ICA with FastICA algorithm to retrieve independent components (ICs) from collected financial time-series data. These ICs represent hidden knowledge of the input data as important features, and the mixing matrix A of these ICs can be utilized for cluster analysis. In the next stage, training data from firms in a cluster are collected to train an SVR model, and the number of SVR models depends on the number of clusters deduced in ICA analysis. The SVR model we consider is the non-linear regression model with cost function involved. Figure 3-1 illustrates the proposed forecasting model. The overall procedure of the proposed approach is shown in Figure 3-2, which is discussed in details as follows.

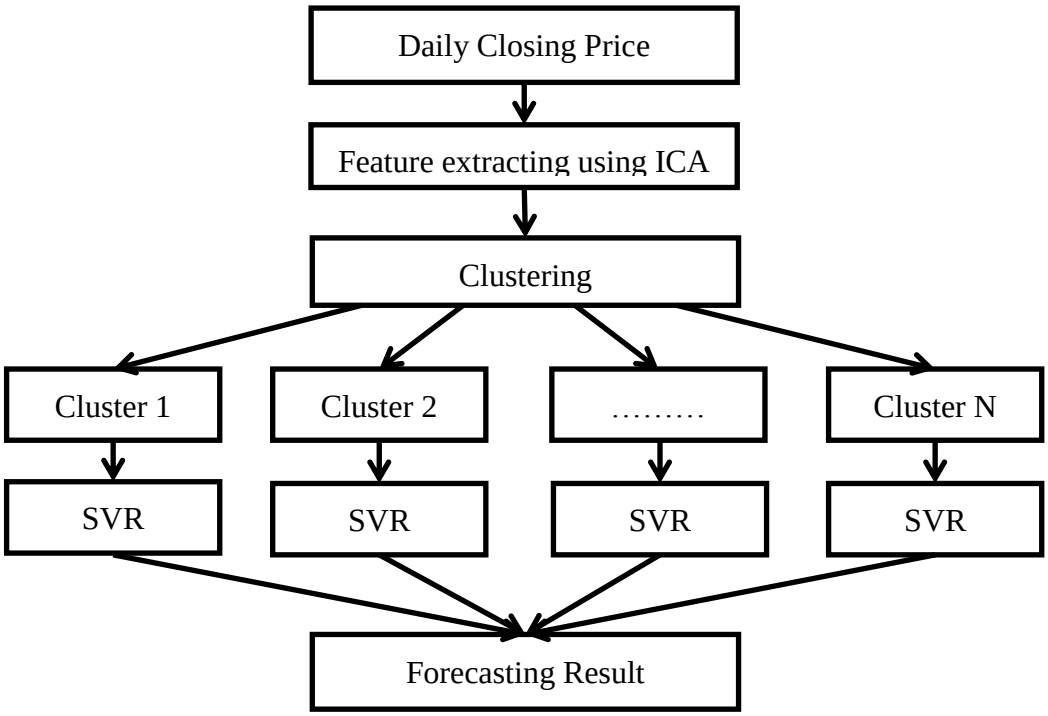


Figure 3-1 The hybrid stock price forecasting model

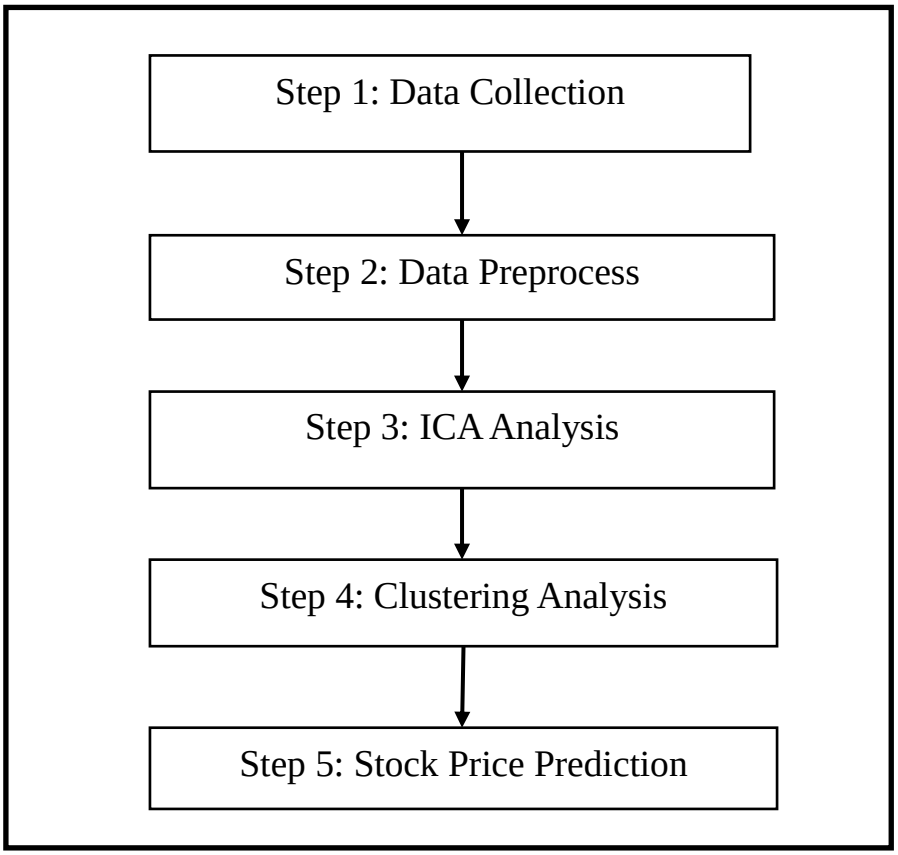


Figure 3-2 Steps of the proposed approach

Step 1 Data Collecting

In this study, we focus on forecasting the stock prices of firms in Taiwan using the ICA technique and the SVR model. The firms chosen are mainly from the *FTSE TWSE Taiwan 50 Index* (*Taiwan 50*), which represent the most highly capitalized blue chip stocks and represent nearly 70% of the Taiwanese market. For each company, the daily closing stock price is recorded from *Yahoo! Finance* (http://finance.yahoo.com/) within a certain time period. The collected data are then fed into the next step for further preprocess.

Step 2 Data Preprocessing

In generally, the raw data collected are not appropriate to use for forecasting purpose directly. Like most financial forecasting techniques, we transform the daily closing price into price change rate, which is defined as

$$\Delta p_i = \frac{p_{i-1}}{p_i} - 1 \qquad (3\text{-}1)$$

where Δp_i represents the differential rate, p_i represents the closing price in the i^{th} day, and p_{i-1} represents the closing price in the $(i-1)^{th}$ day.

After the data transformation, we now perform the preprocessing step of ICA before *FastICA* algorithm is applied. The preprocessing step includes centering, which subtract each data point from its mean value, and whitening, which transform the

observed data vectors of stock prices into uncorrelated ones with variances of unity. Both centering and whitening simplify the ICA algorithm and reduce the number of parameters to be estimated.

Step 3 ICA analysis

In this step, the *FastICA* algorithm is applied to uncover the latent independent components ***s*** from the observed mixing signals ***x***. Their relationship is expressed in Eq. (2-3) and is restated as follows:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \bullet \\ \bullet \\ x_n \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \bullet & \bullet & a_{1n} \\ a_{21} & a_{22} & \bullet & \bullet & a_{2n} \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ a_{n1} & a_{n2} & \bullet & \bullet & a_{nn} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \bullet \\ \bullet \\ s_n \end{pmatrix} = \mathbf{A}\boldsymbol{s} \qquad (3\text{-}2)$$

To solve ***s***, we consider a linear combination of the x_i's, and denote this by $\mathbf{W}^T \boldsymbol{x}$ where **W** is the de-mixing matrix. As described in Section 2.1.3, ***s*** is solvable by maximizing the non- Gaussianity of $\mathbf{W}^T \boldsymbol{x}$. If **W** can be successfully estimated by *FastICA*, then actually its inverse $\mathbf{W}^{-1}$ will reflect the relations between ***x*** and ***s*** and correspond to the mixing matrix **A**.

In real applications, it depends on the situations which matrix, **W** or **A**, is essential to obtain. If the applications are to find the latent independent components, then matrix **W** shows how the ICs are constructed from the linear combination of the observed

signals $\boldsymbol{x}$. On the other hand, if the applications are to make use of ICs for further analysis, e.g. the clustering analysis in our case, then matrix **A** that represents the loadings between ICs and the observed data places an important role for such a purpose (Wu & Yu, 2005).

The following simple example illustrates how the matrix A reflects the latent structure of the observed data. Assume there are three mixing signals, x_1, x_2, and x_3, and we desire to find the ICs, s_1, s_2, and s_3. After applying *FastICA*, matrix **W** is solved and its inverse that corresponds to matrix A shows the relations between $\boldsymbol{x}$ and $\boldsymbol{s}$ as follows:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.8 & 0.2 & 0.09 \\ 0.96 & 0.33 & 0.4 \\ 0.25 & 0.7 & 0.1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} \tag{3-3}$$

or equivalently, using equation expressions:

$$x_1 = 0.8 \times s_1 + 0.2 \times s_2 + 0.09 \times s_3$$

$$x_2 = 0.96 \times s_1 + 0.33 \times s_2 + 0.4 \times s_3$$

$$x_3 = 0.25 \times s_1 + 0.7 \times s_2 + 0.1 \times s_3 \tag{3-4}$$

The element values of matrix **A** are the loadings of s_1, s_2, and s_3. Telling from the loadings, it is clear that s_1 is the most dominant independent component for x_1 and so is it for x_2. In contrast, s_2 is the most dominant independent component for x_3. This analysis shows that x_1 and x_2 are similar while x_3 are quite different from both of them. In terms of cluster analysis, x_1 and x_2 may be grouped together while x_3 needs to stay alone to form a cluster of its own.

The next issue in ICA analysis is to select only essential ICs to remove noise and redundancy in the original signals. We adopt the sub-optimal search approach as proposed by Cheung and Xu, 2000 because it is relatively efficient in computational efforts.

We first define the relative Hamming distance (RHD) between the signals from a combination of selected ICs and the original ones. Let the original i^{th} signal be denoted by x_i and the i^{th} combined signal from selected ICs be $\hat{x}_i$. Then RHD is defined as

$$\text{RHD}(\boldsymbol{x}, \hat{x}) = \frac{1}{n(T-1)} \sum_{i=1}^{n} \sum_{t=1}^{T-1} (R_i(t) - \hat{R}_i(t))^2 \qquad (3\text{-}5)$$

where n is the number of ICs, T is the number of timestamp points, $R_i(t)$ is the sign function of $(x_i(t)-x_i(t+1))$, and $\hat{R}_i(t)$ is the sign function of $(\hat{x}_i(t)-\hat{x}_i(t+1))$.

With RHD defined, the search algorithm is expressed in the following steps:

Step 1: Given a component subscript set $Z = \{j \mid 1 \le j \le n\}$, let k =1, and the order list of ICs, $L = \varnothing$.

Step 2: For each $j \in Z$, let $\hat{x}_{i-j} = \sum_{m \ne j,\, j \in Z} a_{im} s_m$ denote the i^{th} signal constructed from ICs with the corresponding subscripts in Z but excluding j, where a_{im} denotes the corresponding element in matrix **A**. Then select the subscript $\beta = \text{argmin}_{j \in Z}$ $\text{RHD}(\boldsymbol{x}, \hat{x}_{-j})$ as the k^{th} element in L, where $\hat{x}_{-j}$ denotes the constructed signals from ICs with the corresponding subscripts in Z but excluding j. Then let $Z = Z - \{\beta\}$.

Step 3: If Z $\not\subset \varnothing$, then $k = k + 1$, and go to Step 2; otherwise, go to Step 4.

Step 4: The resultant list L denotes the ICs with corresponding subscripts in ascending order (the latter, the more important).

However, with the above approach, we can only order the importance of ICs without knowing how many of them to select. We therefore adopt the empirical rule for the number of ICs as the number of essential eigenvalues in matrix A. The result can be expressed as

$$\boldsymbol{x}' = \mathbf{A}' \boldsymbol{s}' \tag{3-6}$$

where $\boldsymbol{s}'$ denotes the selected IC subset, $\mathbf{A}'$ contains row vectors corresponding to $\boldsymbol{s}'$, and $\boldsymbol{x}'$ is the constructed signals from the select ICs in $\boldsymbol{s}'$.As a result, each loading (row) vector in matrix **A'** reflects the latent structure of the observed mixing signals with unimportant ICs removed, from which the clustering algorithm can be applied to form clusters for the observed mixing signals.

Step 4 Clustering

This step is to perform cluster analysis for the observed stock prices of companies in terms of their IC structure. Given the mixing matrix **A**, the agglomerative hierarchical clustering algorithm is adopted to perform the clustering task. Ward's method is applied to the proximity estimation of the clustering procedure.

Take the simple example described in Step 3. Each x_i is regarded as a cluster initially. Then the merging cost of combining x_1 and x_2, x_2 and x_3, and x_1 and x_3 is shown as

$$\Delta(x_1, x_2) = \frac{1}{2}[(0.8–0.96)^2 + (0.2–0.33)^2 + (0.09–0.4)^2] = 0.278$$

$$\Delta(x_2, x_3) = \frac{1}{2}[(0.96–0.25)^2 + (0.33–0.7)^2 + (0.4–0.1)^2] = 1.462$$

$$\Delta(x_1, x_3) = \frac{1}{2}[(0.8–0.25)^2 + (0.2–0.7)^2 + (0.09–0.1)^2] = 1.104 \qquad (3\text{-}5)$$

Therefore, we should cluster x_1 and x_2 first (as expected) and continue the procedure until all data points are merged into one single cluster.

Step 5 Stock Price Predicting

Finally, in this step, we will apply support vector regression (SVR) to predict the stock prices. As mentioned before, the stock prices of individual firms may exhibit very different trends. Therefore, it makes no sense to predict stock prices of all firms as a whole with a single SVR model. Instead, after the previous step of cluster analysis, firms with similar stock price trends are clustered and it makes more sense to predict stock prices of firms within a cluster with a single SVR model. The number of SVR models is contingent upon the number of clusters found in Step 4.

The issue in this step becomes how we manipulate the data into the SVR model. The original time-series financial data do not fit into the model where inputs and

outputs are required. Therefore, we need to extract the inputs and outputs from the existing collected data.

Tay and Cao (Tay & Cao, 2001) provided the ideas of transforming the original data into 5 input variables (MA15, RDP−5, RDP−10, RDP−15, RDP−20) and 1 output variable (RDP+5). The definition of these variables is shown in Table 3-1. In our study, we simply follow this definition to build the SVR models.

Table 3-1 Definition of input and output variables in SVR

Indicator	Definition
Input variables	
MA15	$p_i - \overline{MA15_i}$
RDP-5	$\frac{(p_i - p_{i-5})}{p_{i-5}} \times 100$
RDP-10	$\frac{(p_i - p_{i-10})}{p_{i-10}} \times 100$
RDP-15	$\frac{(p_i - p_{i-15})}{p_{i-15}} \times 100$
RDP-20	$\frac{(p_i - p_{i-20})}{p_{i-20}} \times 100$
Output variable	
RDP+5	$\frac{(p_{i+5} - p_i)}{p_i} \times 100,\ p_i = \overline{MA3_i}$
Notice: $\overline{MA15_i}$ is the 15-day moving average of the i^{th} day; p_i of each input variables are the i^{th} day closing price; and p_i of output variable is defined as $\overline{MA3_i}$, the 3-day moving average of the i^{th} day.	

Chapter 4 Experiments and Results

In this chapter, we conduct two experiments to examine the performance of our proposed hybrid approach to predicting stock prices as described in Chapter 3. Particularly, we compare the performance between our proposed approach and the one that treats all companies as a whole with a single SVR prediction model.

In the following, we will discuss the experimental design employed in the experiments. Experiment results are then presented and examined. Through these experiments, we desire to demonstrate the feasibility of our approach to enhance the stock price forecasting performance.

4.1 Experimental Design

This section describes how we conduct experiments to validate our proposed approach, which includes data collection, data transformation, objectives of the experiments, performance measure, and evaluation scheme.

Data Collection

In this study, the firms we focus on are mainly from the *FTSE TWSE Taiwan 50 Index* (*Taiwan 50*). Companies listed in *Taiwan 50* are the most highly capitalized blue chip stocks and represent nearly 70% of the Taiwanese market. For each company, the daily closing stock price is recorded and collected from *Yahoo! Finance* (http://finance.yahoo.com/) with a time period from 2009/8/18 to 2010/10/27. However, during this time period, some of these companies contain too many missing values or

contain highly fluctuating daily price due to unknown external factors. For example, *Asustek Computer Inc*. had been delisted in the period from 2010/5/17 to 2010/6/24. The disappeared daily closing prices can be hardly estimated and the deviation of closing prices before re-listed can be hardly smoothed. Therefore, such company data will not be considered in our study. After screening, we choose 40 companies out of *Taiwan 50* for further analysis. The companies are listed in Table 4-1. We therefore collect a total of 300 raw daily closing prices for each of the 40 companies.

Data Transformation

As stated in Chapter 3, the collected raw data needs to be preprocessed to facilitate the analysis. We first transform the raw data into the price change rate as defined in eq. (3-1). For example, the daily closing stock prices within the data collection time period for *Taiwan Semiconductor Manufacturing Company* (*TSMC*) are shown in Figure 4-1, while the transformed results are shown in Figure 4-2. The associated statistics of the raw data and the transformed data are listed in Table 4-2.

Table 4-1 Companies under study

NO.	Companies	Issued shares	weight of TW50	NO.	Companies	Issued shares	weight of TW50
1101	臺灣水泥	3,692,175,869	1.29%	**2353**	宏碁	2,834,726,828	0.76%
1102	亞洲水泥	3,135,817,148	0.86%	**2354**	鴻準	1,172,719,981	1.24%
1216	統一企業	4,544,368,665	2.15%	**2382**	廣達	3,832,574,432	2.11%
1301	臺灣塑膠	6,120,839,632	4.55%	**2474**	可成	750,691,371	1.25%
1303	南亞塑膠	7,852,298,603	4.04%	**2498**	宏達電	852,052,170	2.35%
1326	臺灣化纖	5,690,472,133	3.09%	**2801**	彰銀	5,242,328,404	0.60%
1402	遠東新	4,897,217,356	1.16%	**2880**	華南金	8,214,314,422	0.98%
1722	台灣肥料	980,000,000	0.45%	**2881**	富邦金	8,957,747,952	1.98%
2002	中國鋼鐵	15,046,209,352	2.92%	**2882**	國泰金	10,432,471,890	2.18%
2105	正新橡膠	2,472,474,959	1.37%	**2883**	開發金	15,175,533,631	1.02%
2201	裕隆汽車	1,560,340,573	0.55%	**2885**	元大金	10,016,140,025	1.31%
2207	和泰汽車	546,179,184	0.73%	**2886**	兆豐金	11,392,838,223	1.88%
2301	光寶科	2,296,218,012	0.84%	**2891**	中信金	11,412,707,874	1.90%
2303	聯電	12,987,771,315	1.51%	**2892**	第一金	7,665,434,651	0.98%
2308	台達電	2,383,486,207	2.13%	**2912**	統一超商	1,039,622,256	1.18%
2311	日月光	6,712,086,477	1.53%	**3008**	大立光	134,140,197	0.56%
2324	仁寶	4,420,951,525	1.16%	**3045**	台哥大	3,420,832,827	1.53%
2325	矽品	3,116,361,139	0.85%	**3231**	緯創	2,052,140,146	0.70%
2330	臺積電	25,753,417,412	18.71%	**4904**	遠傳	3,258,500,810	1.09%
2347	聯強	1,561,063,559	1.05%	**6505**	台塑化	9,486,083,651	1.52%

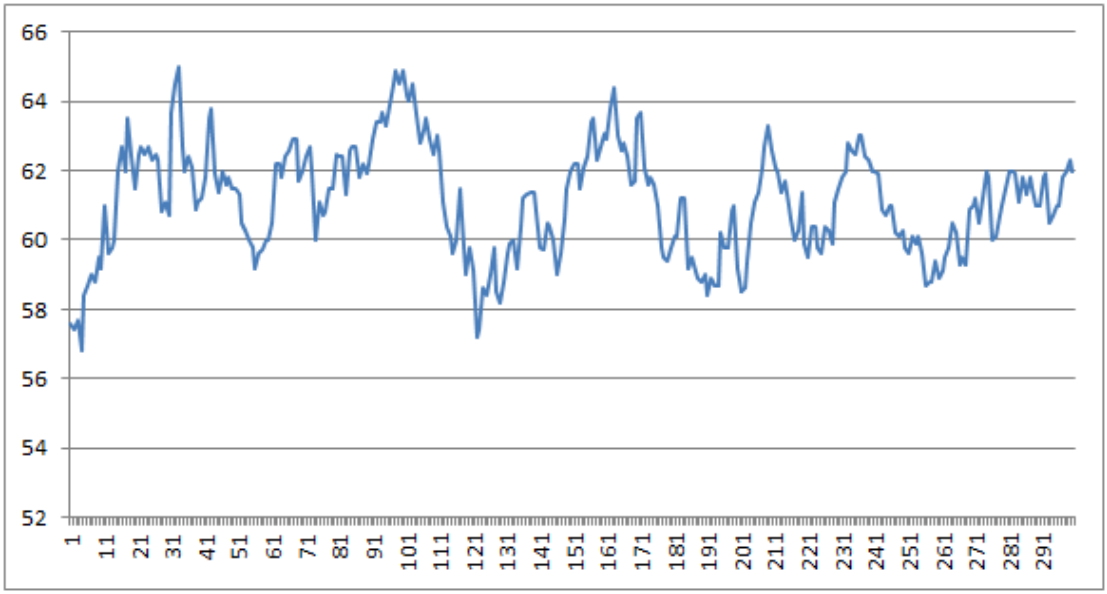


Figure 4-1 Daily closing stock prices for TSMC

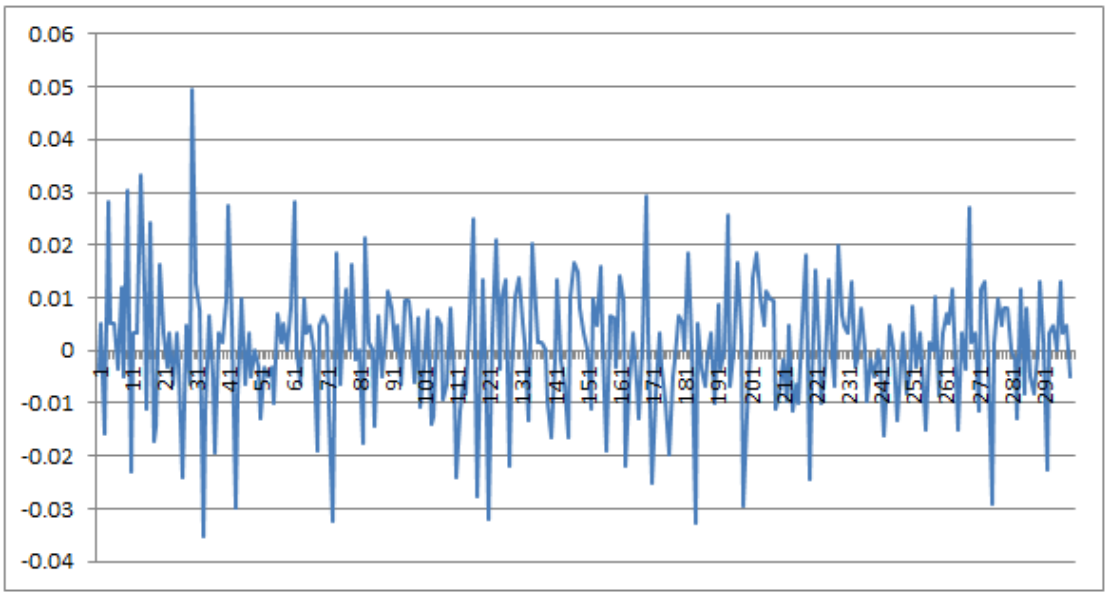


Figure 4-2 Stock price change rates for TSMC

Table 4-2 Statistics for TSMC data

Data type	**Mean**	**Standard deviation**	**Minimum**	**Maximum**
Daily Price	58.833	5.937	39.6	72.9
Volatility	0.0009	0.016	-0.067	0.069

Objectives of the Experiments

To justify the hybrid approach to predicting stock prices, we conduct two experiments. The first experiment is designed to examine how our proposed approach performs and compare with the performance of a single SVR model that treats all companies as a whole without clustering. The second experiment is conducted after

further investigating the clustering results by removing outliers. Both experimental results can validate the feasibility of applying cluster analysis in stock price forecasting.

Table 4-3 Objectives of the Experiments

	Objectives
Experiment I	To examine the performance of our proposed approach.
Experiment II	To further refine the cluster analysis and examine the performance of our proposed approach.

Performance Measure

Our study adopts *root mean squared error* (RMSE) to measure the stock price forecasting performance. RMSE is a common measure to apply in prediction problems. It is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2} \quad (4\text{-}1)$$

where $f(x_i)$ is the predicted value, and y_i is the actual value. The smaller the values of RMSE, the closer are the predicted time series values to the actual values.

Evaluation Scheme

Recall from Step 5 in Chapter 3 that we will transform the time-series data into data of 5 input variables and 1 output variable to fit SVR requirements. With those input-output data ready, we adopt the evaluation scheme of hold-out validation in our

experiments, which randomly splits the data into a training set and a test set according to 70–30 percentages. The training data are used to train the SVR model and the test data are used to examine how the trained SVR model performs. Finally, to make the performance results more robust, we partition the data according to their corresponding time period. Time period I refers to the duration from 2009/8/18 to 2010/1/6, time period II refers to the duration from 2010/1/7 to 20106/7, and time period III refers to the duration from 2010/6/8 to 2010/10/27. Therefore, the experiments are repeated under the three time periods to obtain results of three replications.

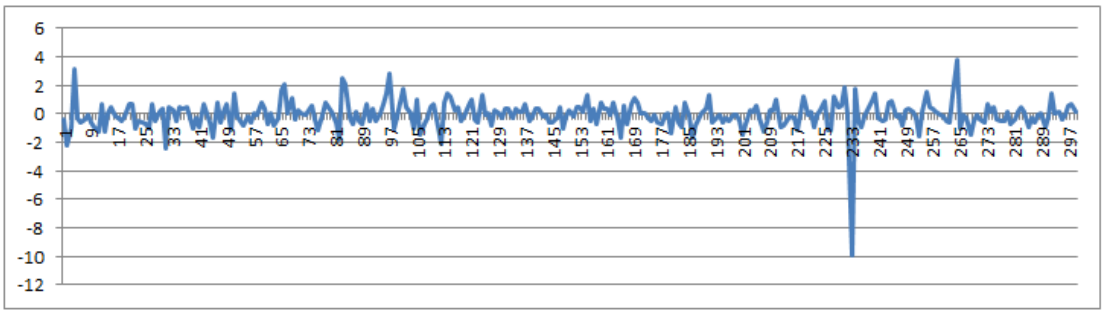
4.2 Experiment I

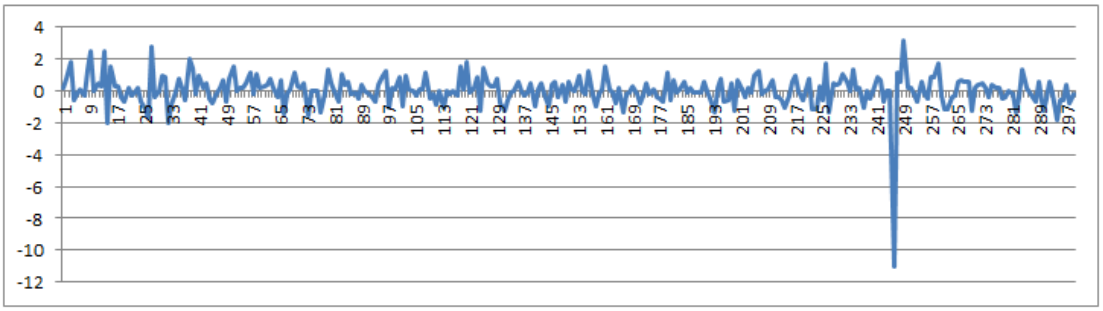
The objective of Experiment I is to examine how our proposed approach performs, which basically includes clustering analysis and SVR model training to predict the stock prices. The tasks in this experiment included SVR prediction based on clustering with all ICs, clustering with selected ICs, and without clustering.

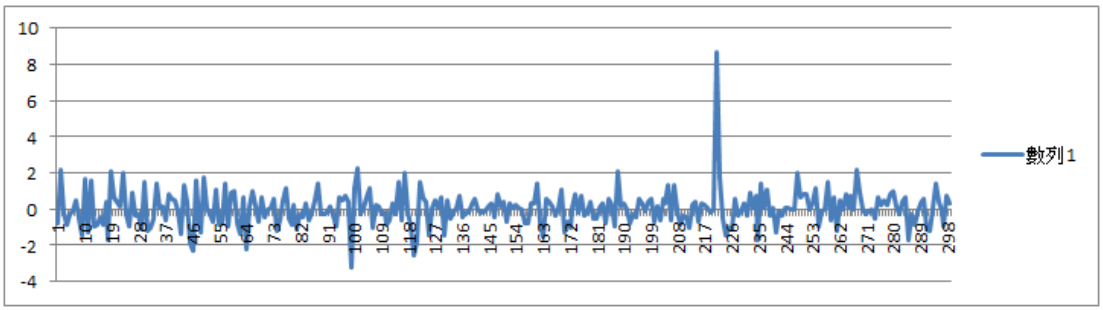
SVR prediction based on clustering with all ICs

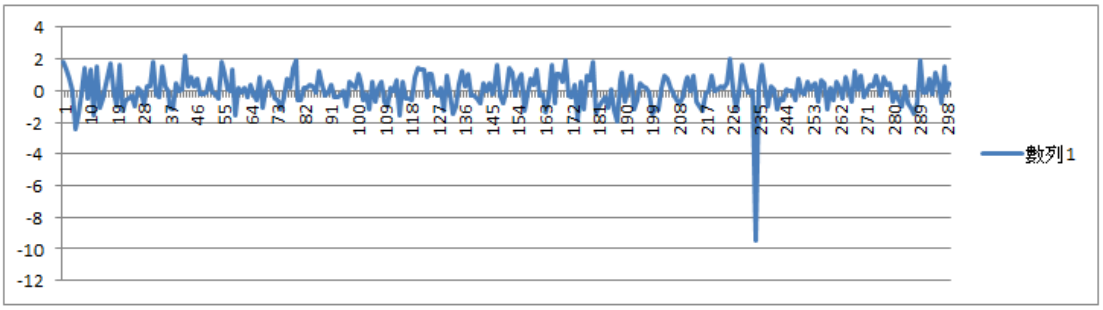
We first apply the ICA analysis to the collected data to extract their latent structure with the *FastICA* algorithm, which is implemented in a java library. In this case, the number of ICs obtained is equal to the number of companies we select. Figure 4-3

exhibits the first 6 ICs in our study. Notice that unlike Principal Component Analysis (PCA), the order of the ICs does not represent their degree of importance. The ICs uncovered only refers to underlying factors that may affect the stock prices.









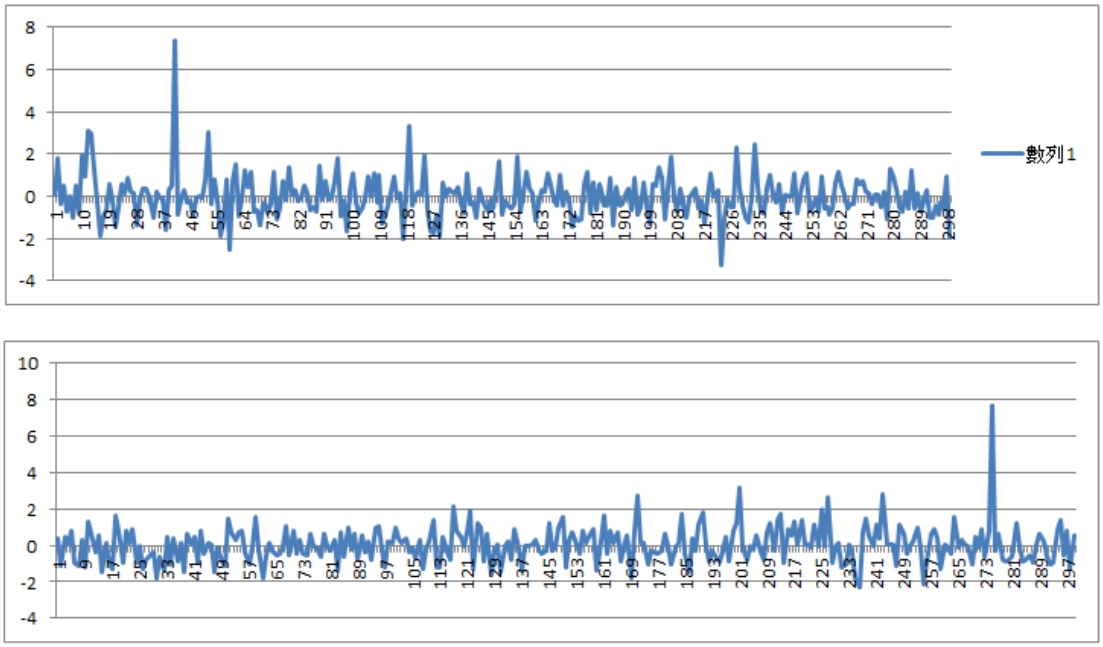


Figure 4-3 The first 6 ICs in our collected data

Furthermore, de-mixing matrix **W** is also generated with the *FastICA* algorithm. And its inverse is the mixing matrix **A**, whose loadings reflects the relationships between the observed data and ICs. The agglomerative hierarchical clustering technique together with Ward's distance measure is then applied to loadings in **A** for the clustering task. Figure 4-4 shows the dendrogram of the clustering results where Clusters C_1 to C_{40} represents the 40 companies. After choosing an appropriate cut-point in the dendrogram, five clusters results accordingly. The cluster results of the companies are listed in Table 4-4.

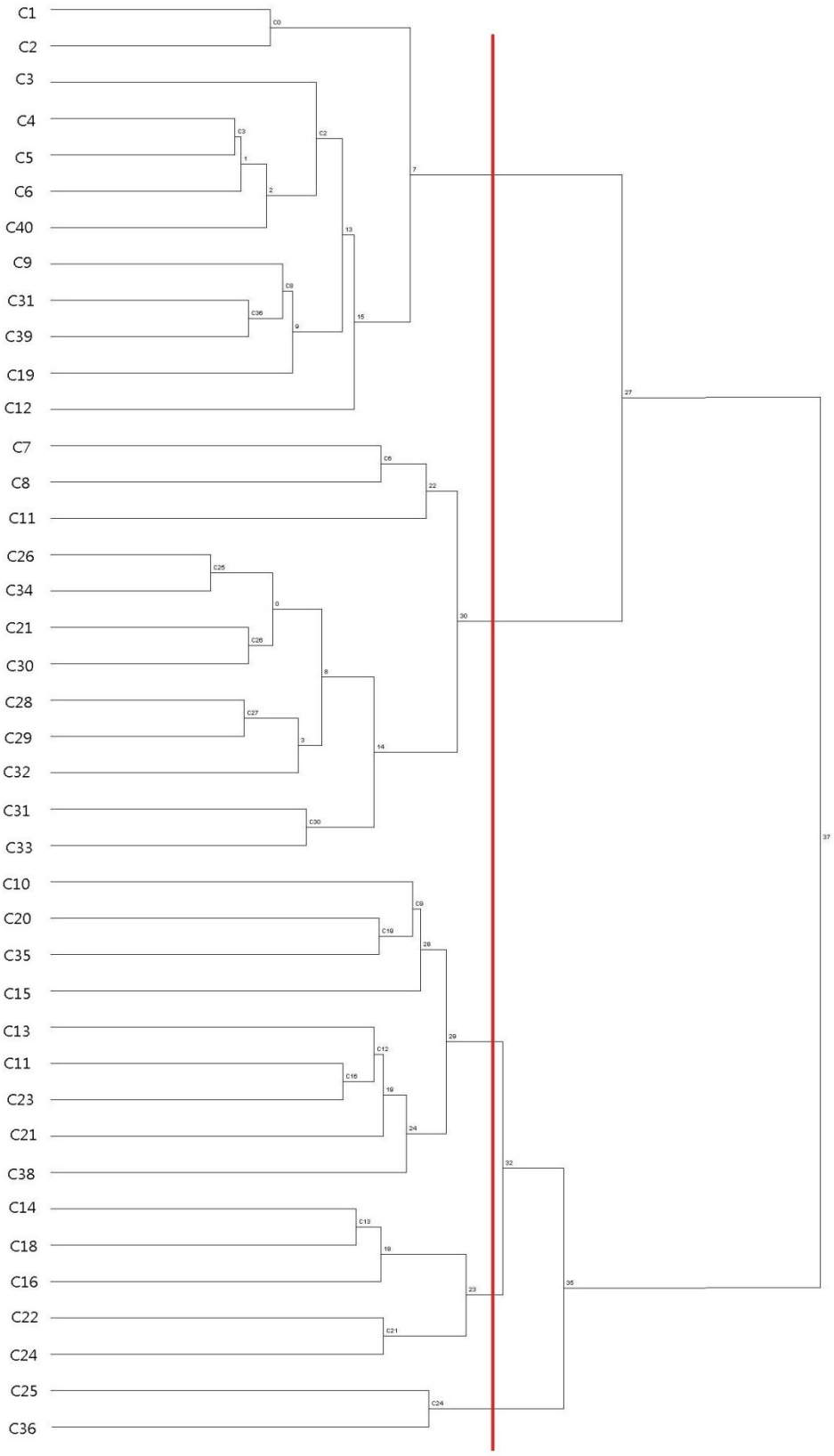


Figure 4-4 Dendrogram of the clustering results based on all ICs

Table 4-4 Cluster results of the companies

Cluster 1 (12 companies)				**Cluster 2 (12 companies)**			
1101	臺灣水泥	**2002**	中國鋼鐵	**1402**	遠東新	**2882**	國泰金
1102	亞洲水泥	**2207**	和泰汽車	**1722**	台灣肥料	**2883**	開發金
1216	統一企業	**2330**	臺積電	**2201**	裕隆汽車	**2885**	元大金
1301	臺灣塑膠	**2474**	台哥大	**2801**	彰銀	**2886**	兆豐金
1303	南亞塑膠	**2498**	遠傳	**2880**	華南金	**2891**	中信金
1326	臺灣化纖	**2801**	台塑化	**2881**	富邦金	**2892**	第一金
Cluster 3 (9 companies)				**Cluster 4 (5 companies)**			
2105	正新橡膠	**2353**	宏碁	**2303**	聯電	**2354**	鴻準
2301	光寶科	**2382**	廣達	**2311**	日月光	**2474**	可成
2308	台達電	**2912**	統一超商	**2325**	矽品		
2324	仁寶	**3231**	緯創	**Cluster 5 (2 companies)**			
2347	聯強			**2498**	宏達電	**3008**	大立光

The next step is to build SVR models for each cluster listed in Table 4-4. As mentioned, we partition the data according to their time periods into three groups and repeat the SVR modeling with each group of data. Therefore, three replications are done in this experiment. Each group of data contains 100 daily closing prices for each individual company. Furthermore, to transform those data into the input-output format employed in SVR, we will obtain only 75 input-output paired data among the 100 closing prices. This is because we adopt the input variable of RDP−20 and the output variable of RDP+5, defined in Table 3.1, so that the first 20 data and the last 5 data are ignored. Then 70% of the 75 paired data for each individual company are used for training and 30% of them are for test.

The last step is to train the SVR models based on the training data while adjust the models based on the test data. To adjust an SVR model is equivalent to adjust the parameter values employed, which includes ε, C, and γ. The parameter ε is involved in the ε-intensive loss function that affects the selection of support vectors to construct the regression function. A large ε value will result in less number of support vectors and consequently in a less complex regression function, which may not be always applicable. In our case, we search ε in the range of 2^{-9} to 2^{-1}.

The parameter C denotes the misclassification cost of the model. The choice of the **C** value influences the significance of the individual data points in the training set. According to the SVR formulation, **C** can takes values between 0 and ∞. In out study, we vary **C** between 2^{-15} to 2^{15}. Finally, the parameter γ denotes the radius used in the exponential radial basis function. We again vary γ between 2^{-15} to 2^{15} to find the optimal kernel radius. Table 4-5, 4-6, and 4-7 lists the forecasting performance for companies in each cluster at each time period.

Table 4-5 Forecasting performance at time period I based on clustering with all ICs

Time period I				
2009/8/18 to 2010/1/6	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	2.9804	2^{9}	2^{-1}	2^{-7}
Cluster 2	3.2685	2^{0}	2^{-4}	2^{-7}
Cluster 3	2.8992	2^{7}	2^{-5}	2^{-9}
Cluster 4	3.6176	2^{11}	2^{-3}	2^{-14}
Cluster 5	3.3461	2^{0}	2^{-1}	2^{1}

Table 4-6 Forecasting performance at time period II based on clustering with all ICs

Time period II				
2010/1/7 to 20106/7	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	2.8437	2^{-2}	2^{-1}	2^{-3}
Cluster 2	3.389	2^{-4}	2^{-8}	2^{-15}
Cluster 3	3.8184	2^{-2}	2^{-9}	2^{-1}
Cluster 4	4.6108	2^{2}	2^{-8}	2^{-1}
Cluster 5	4.3288	2^{4}	2^{-9}	2^{-4}

Table 4-7 Forecasting performance at time period III based on clustering with all ICs

Time period III				
2010/6/8 to 2010/10/27	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	3.4278	2^{7}	2^{-6}	2^{-2}
Cluster 2	3.2606	2^{8}	2^{-4}	2^{-6}
Cluster 3	4.115	2^{-1}	2^{-9}	2^{-5}
Cluster 4	3.7372	2^{0}	2^{-2}	2^{-15}
Cluster 5	5.2829	2^{4}	2^{-1}	2^{-10}

SVR prediction based on clustering with selected ICs

As stated, not all ICs are essential to the original time-series data. Therefore, in this task, we analyze the importance of eigenvalues in matrix **A** to determine the number of ICs to be 19, and adopt the sub-optimal search approach to ordering the ICs. Finally, the subscript set of the chosen ICs is {36, 20, 35, 38, 18, 15, 26, 10, 8, 1, 34, 32, 5, 3, 37, 30, 9, 23, 33} in descending order (the more forward, the more important).

With the chosen 19 ICs, we reapply the agglomerative hierarchical clustering technique together with Ward's distance measure to loadings in **A'** (corresponding to the chosen ICs) for the clustering task. Figure 4-5 shows the dendrogram of the clustering results where Clusters C_1 to C_{40} represents the 40 companies. After choosing an appropriate cut-point in the dendrogram, we obtain five clusters accordingly. Note that even though once again there are five clusters, this clustering result is different from what we obtain based on all ICs (compared with Figure 4-4).

The next step is to build SVR models for each cluster. Table 4-8, 4-9, and 4-10 lists the forecasting performance for companies in each cluster at each time period.

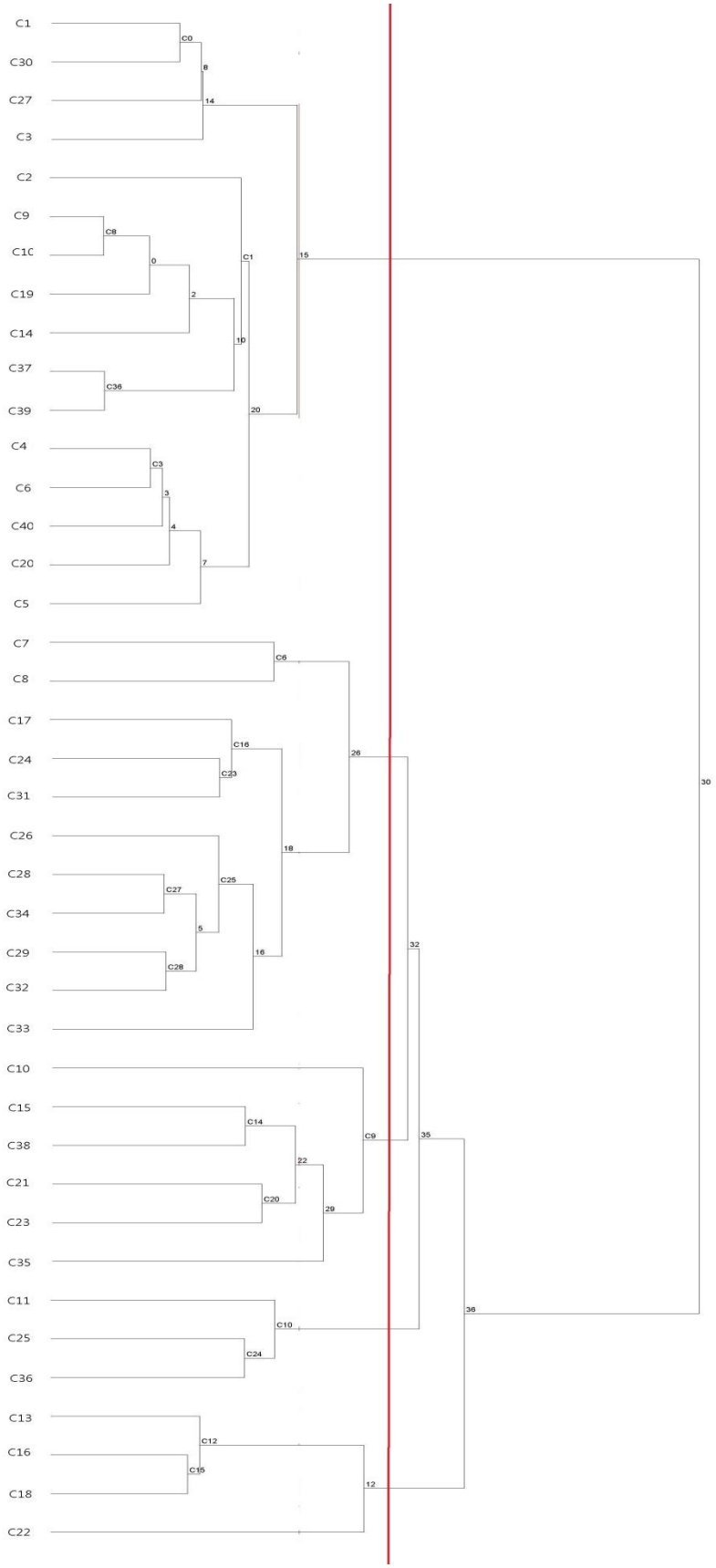


Figure 4-5 Dendrogram of the clustering results based on 19 ICs

Table 4-8 Forecasting performance at time period I based on clustering with 19 ICs

Time period I				
2009/8/18 to 2010/1/6	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	2.744	2^{7}	2^{-1}	2^{-14}
Cluster 2	3.268	2^{1}	2^{-4}	2^{-8}
Cluster 3	2.32	2^{-1}	2^{-3}	2^{1}
Cluster 4	3.5524	2^{1}	2^{-9}	2^{-3}
Cluster 5	4.78	2^{-7}	2^{-4}	2^{-6}

Table 4-9 Forecasting performance at time period II based on clustering with 19 ICs

Time period II				
2010/1/7 to 20106/7	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	3.4014	2^{-3}	2^{-9}	2^{-6}
Cluster 2	2.7331	2^{12}	2^{-2}	2^{-12}
Cluster 3	3.1953	2^{3}	2^{-14}	2^{-10}
Cluster 4	3.9824	2^{6}	2^{-9}	2^{-14}
Cluster 5	4.2332	2^{1}	2^{-3}	2^{-8}

Table 4-10 Forecasting performance at time period III based on clustering with 19 ICs

Time period III				
2010/6/8 to 2010/10/27	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	3.0626	2^{-7}	2^{-1}	2^{3}
Cluster 2	2.8513	2^{-3}	2^{-9}	2^{-7}
Cluster 3	4.3588	2^{-1}	2^{-1}	2^{-7}
Cluster 4	5.2867	2^{10}	2^{-9}	2^{-15}
Cluster 5	4.8723	2^{0}	2^{-9}	2^{-6}

SVR prediction without clustering

The task is to build SVR models without considering clustering for performance comparison purpose. Table 4-11 lists the forecasting performance for companies at each time period.

Table 4-11 Forecasting performance without clustering

Time Period I			
Parameter C	**Parameter ε**	**Parameter γ**	**RMSE**
2^{6}	2^{-3}	2^{-6}	3.204
Time Period II			
2^{5}	2^{-9}	2^{-10}	2.9412
Time Period III			
2^{2}	2^{-4}	2^{-7}	3.1272

Performance comparison

Finally, we desire to compare the performance with ICA clustering and without clustering. For a fair comparison, we need to take weighted average of the RMSEs with clustering results to be a single measure. For example, at time period I, the weighted RMSE based on clustering with all ICs is calculated as

$$2.9804 \times \frac{12}{40} + 3.2685 \times \frac{12}{40} + 2.8992 \times \frac{5}{40} + 3.6176 \times \frac{9}{40} + 3.3461 \times \frac{2}{40} = 3.1465$$

Table 4-12 shows all the performance results for the comparison purpose.

Table 4-12 Performance comparison in experiment I

	RMSE		
	Time Period I	**Time Period II**	**Time Period III**
ICA(40) + SVR	3.1465	3.5217	3.6637
ICA(19) + SVR	3.0847	3.3135	3.5467
SVR	3.204	2.9412	3.1272

We first observe from Table 4-12 that the prediction results based on clustering with 19 ICs are better than those based on clustering with all ICs. Therefore, it justifies our conjecture that time-series data constructed from selected essential ICs are less noisy and redundant, and help in the prediction performance.

However, it is to our surprise that the prediction results without clustering are better in Time Period II and Time Period III than those based on clustering with all ICs and clustering with 19 ICs, which is contradictory to our intuition. We then look at the clustering result in Figure 4-4, and find out that the final cluster containing only C25 (2498, 宏達電) and C36 (3008, 大立光) contribute high RMSE. Both of them have much higher stock prices within the time period and exhibit different patterns from the rest companies. They are considered as outliers in the clustering analysis and may even deteriorate the prediction performance by solely focusing on them. With such an observation, we conduct further experiments to examine how our proposed approach performs without considering these two companies.

4.3 Experiment II

Experiment II is to reexamine our proposed approach without considering two outlier companies. Again, the tasks in this experiment included SVR prediction based on clustering with all ICs, clustering with selected ICs, and without clustering.

SVR prediction based on clustering with all ICs

We actually utilize the cluster result in Figure 4-4 and simply remove the final cluster from consideration. Therefore, there are totally four clusters in this task. The next step is to build SVR models for each cluster. Table 4-13, 4-14, and 4-15 lists the forecasting performance for companies in each cluster at each time period.

Table 4-13 Forecasting performance of 38 companies at time period I based on clustering with all ICs

Time period I				
2009/8/18 to 2010/1/6	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	2.9804	2^{9}	2^{-1}	2^{-7}
Cluster 2	3.2685	2^{0}	2^{-4}	2^{-7}
Cluster 3	2.8992	2^{7}	2^{-5}	2^{-9}
Cluster 4	3.6176	2^{11}	2^{-3}	2^{-14}

Table 4-14 Forecasting performance of 38 companies at time period II based on clustering with all ICs

Time period II				
2010/1/7 to 20106/7	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	2.8437	2^{-2}	2^{-1}	2^{-3}
Cluster 2	3.389	2^{-4}	2^{-8}	2^{-15}
Cluster 3	3.8184	2^{-2}	2^{-9}	2^{-1}
Cluster 4	4.6108	2^{2}	2^{-8}	2^{-1}

Table 4-15 Forecasting performance of 38 companies at time period III based on clustering with all ICs

Time period III				
2010/6/8 to 2010/10/27	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	3.4278	2^{7}	2^{-6}	2^{-2}
Cluster 2	3.2606	2^{8}	2^{-4}	2^{-6}
Cluster 3	4.115	2^{-1}	2^{-9}	2^{-5}
Cluster 4	3.7372	2^{0}	2^{-2}	2^{-15}

SVR prediction based on clustering with selected ICs

The task is to build SVR models based on clustering with the chosen 19 ICs. The subscript set of the chosen ICs in this case is {32, 34, 36, 25, 18, 26, 12, 33, 35, 27, 14, 22, 11, 40, 21, 6, 15, 28, 20, 38} in descending order. Figure 4-6 shows the dendrogram of the clustering results. After choosing an appropriate cut-point in the dendrogram, we again obtain four clusters accordingly.

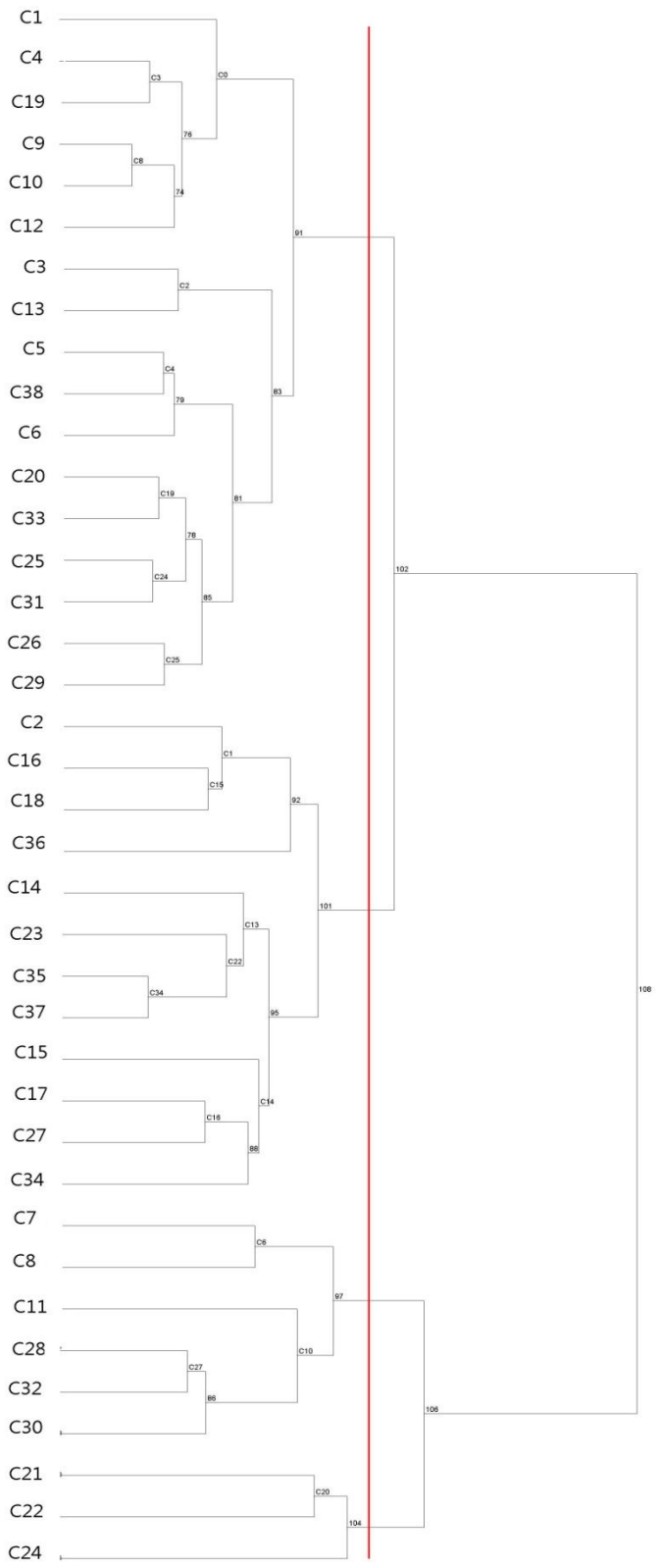


Figure 4-6 Dendrogram of the clustering results based on 19 ICs in 38 companies

The next step is to build SVR models for each cluster. Table 4-16, 4-17, and 4-18 lists the forecasting performance for companies in each cluster at each time period.

Table 4-16 Forecasting performance of 38 companies at time period I based on clustering with 19 ICs

Time period I				
2009/8/18 to 2010/1/6	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	2.9858	2^{6}	2^{-2}	2^{-12}
Cluster 2	3.1181	2^{-3}	2^{-9}	2^{-15}
Cluster 3	3.5441	2^{1}	2^{-1}	2^{-6}
Cluster 4	2.9063	2^{-2}	2^{-9}	2^{-3}

Table 4-17 Forecasting performance of 38 companies at time period II based on clustering with 19 ICs

Time period II				
2010/1/7 to 20106/7	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	2.6172	2^{1}	2^{-3}	2^{-6}
Cluster 2	3.1921	2^{7}	2^{-9}	2^{-12}
Cluster 3	3.3911	2^{2}	2^{-3}	2^{-5}
Cluster 4	5.1613	2^{1}	2^{-9}	2^{-5}

Table 4-18 Forecasting performance of 38 companies at time period III based on clustering with 19 ICs

Time period III				
2010/6/8 to 2010/10/27	**RMSE**	**Parameter C**	**Parameter ε**	**Parameter γ**
Cluster 1	3.061	2^{-3}	2^{-9}	2^{-7}
Cluster 2	3.6253	2^{-4}	2^{-8}	2^{-7}
Cluster 3	2.9427	2^{1}	2^{-9}	2^{2}
Cluster 4	3.36	2^{7}	2^{-6}	2^{-10}

SVR prediction without clustering

The task is to build SVR models without considering clustering for performance comparison purpose. Table 4-19 lists the forecasting performance for companies at each time period.

Table 4-19 Forecasting performance without clustering

Time Period I			
Parameter C	**Parameter ε**	**Parameter γ**	**RMSE**
2^{-10}	2^{-4}	2^{-7}	3.0974
Time Period II			
2^{10}	2^{-7}	2^{-12}	2.9171
Time Period III			
2^{12}	2^{-5}	2^{-8}	2.9838

Performance comparison

Finally, we desire to compare the performance with ICA clustering and without

clustering. Table 4-20 shows all the performance results for the comparison purpose.

Table 4-20 Performance comparison in experiment II

	RMSE		
	Time Period I	**Time Period II**	**Time Period III**
ICA(40) + SVR	2.9792	3.3053	3.3995
ICA(19) + SVR	2.8014	2.8205	2.9148
SVR	3.0974	2.9171	2.9838

We first observe from Table 4-20 that the prediction results based on clustering with 19 ICs are again better than those based on clustering with all ICs. Therefore, selecting essential ICs does help in the prediction performance.

Next, we compare the prediction results with clustering to those without clustering. Again, the performance without clustering still perform quite well and even better than that with clustering based on 40 ICs. However, with ICs carefully selected, the prediction performance can be enhanced to even outperform the performance without clustering, which complies our underlying thinking that it is better to predict the stock prices for firms in the same cluster with a single model, and develop different models for different clusters. The feasibility of our proposed hybrid approach to applying ICA in cluster analysis and incorporating SVR for stock price forecasting is thus justified.

Chapter 5 Conclusion

5.1 Concluding remarks

Financial time-series forecasting is an interesting but fully challenging research issue due to the dynamics of data characteristics in nature. In the past, artificial neural networks are developed to deal with such problems. However, the lacking of model transparency and the loss of generalization ability often makes the forecasting models weak in explanatory power and erroneous in prediction accuracy.

Therefore, in this research, we propose a hybrid model for stock price forecasting. Independent component analysis (ICA) is employed to uncover the latent structure of the observed time series. The extracted components further assist us in clustering the stock price trends of the companies under study. In addition, for time-series data of companies in each individual cluster, we establish an SVR model that is conceptually and empirically better than the traditional forecasting models in terms of generalization ability.

Two experiments are conducted accordingly. The first experiment shows the performance of our proposed approach and compares our results to those from one single trained SVR model. It is observed that the prediction results based on clustering with 19 ICs are better than those based on clustering with all ICs, but both fail to outperform

those without clustering. With careful examine, the reason may lie in the two outlier companies identified in the cluster result. The second experiment is then to reexamine our proposed approach without considering the outliers. The results show that with ICs carefully selected, the prediction performance can be enhanced and outperform the performance without clustering. Although the proposed approach seems promising in financial time-series forecasting, there are an issue that need to be addressed. First, the IC selection approach we adopt is greedy and sub-optimal, even though it is efficient considering the computational efforts. Therefore, an essential future work is to develop a more feasible approach to selecting ICs that guarantee optimality without much time consumption.

Second, in our experiments, the parameter settings of ε, C, and γ in SVR are ad hoc by trial and error. Optimization techniques such as particle swarm optimization (PSO) can be considered to apply so that the performance of SVR forecasting can be enhanced in the optimal way.

REFERENCES

Back, A. D., & Weigend, A. S. (1997). A first application of independent component analysis to extracting structure from stock returns.

Cardoso, J. F. (1998). Blind signal separation: statistical principles. *Proceedings of the IEEE, 86*(10), 2009-2025.

Cheung, Y., & Xu, L. (2001). Independent component ordering in ICA time series analysis. *Neurocomputing, 41*(1), 145-152.

Chien, J. T., & Chen, B. C. (2006). A new independent component analysis for speech recognition and separation. *Audio, Speech, and Language Processing, IEEE Transactions on, 14*(4), 1245-1254.

Hyvärinen, A., & Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural computation, 9*(7), 1483-1492.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks, 13*(4-5), 411-430. doi: 10.1016/s0893-6080(00)00026-5

Le Borgne, H., Guérin-Dugué, A., & Antoniadis, A. (2004). Representation of images for classification with independent features. *Pattern Recognition Letters, 25*(2), 141-154.

Lin, C. T., Ko, L. W., Chung, I. F., Huang, T. Y., Chen, Y. C., Jung, T. P., & Liang, S. F. (2006). Adaptive EEG-based alertness estimation system by using ICA-based fuzzy neural networks. *Circuits and Systems I: Regular Papers, IEEE Transactions on, 53*(11), 2469-2476.

Lin, C. T., Wu, R. C., Liang, S. F., Chao, W. H., Chen, Y. J., & Jung, T. P. (2005). EEG-based drowsiness estimation for safety driving using independent component analysis. *Circuits and Systems I: Regular Papers, IEEE Transactions on, 52*(12), 2726-2738.

Lu, C. J., & Wang, Y. W. (2010). Combining independent component analysis and growing hierarchical self-organizing maps with support vector regression in product demand forecasting. *International Journal of Production Economics, 128*(2), 603-613.

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*.

Naik, G. R., & Kumar, D. K. (2011). An Overview of Independent Component Analysis and Its Applications. *Informatica: An International Journal of Computing and Informatics, 35*(1), 63-81.

Smola, A., & Vapnik, V. (1997). Support vector regression machines. *Advances in*

neural information processing systems, 9, 155-161.

Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining*: Pearson Addison Wesley Boston.

Tay, F., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *OMEGA-OXFORD-PERGAMON PRESS-, 29*, 309-317.

Vapnik, V. N. (1995). The nature of statistical learning theory.

Wu, E., & Yu, P. (2005). Independent component analysis for clustering multivariate time series data. *Advanced Data Mining and Applications*, 730-730.