INTELLIGENT WEB-BASED AGENT SYSTEM (iWAF) FOR e-FINANCE APPLICATION

KWONG WAI MAN, RAYMOND

DEPARTMENT OF COMPUTING
THE HONG KONG POLYTECHNIC UNIVERSITY
HUNG HOM, HONG KONG

APRIL 2004

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF PHILOSOPHY

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma except where due acknowledgement has been made in the text.

_______________________ (Signed)

Kwong Wai Man, Raymond (Name of student)

Abstract

Intelligent Web-Based Agent System (iWAF) for e-Finance Application

E-Finance is a complex challenge, requiring complex strategy development and technology implementation. AI techniques applied to stock market prediction include: expert system, fuzzy logic, neural networks, genetic algorithms and some statistical models. But neither one can give confidence for investors to rely on due to the lack of unbiased market movement analysis, trend prediction, human behaviour and psychological implication studies. In this study, we propose a multi-agent framework which combines all the advantages of different forecasting methods, for predicting the best timing of stock market for investment.

The system contains five forecasting agents using fundamental, technical and statistical analyses to forecast the market. Each agent is a domain expert in a particular forecasting method and they work together to fill the knowledge gap. As the proposed framework contains different forecasting techniques, we anticipate that the system can provide different measure of the financial market and allow investors to position their investment better and make it more effective.

The proposed system consists of five forecasting agents and two non-forecasting agents. Forecasting agents include Fundamental Agent, Technical Agent, Associate Agent, Adaptive Agent and Expert Agent. Different agents give different recommendations in the investment process. Furthermore, we just focus on the construct of 3 agents which are the Coordinate Agent, Technical Agent and Adaptive Agent.

The Coordinate Agent is to collect all the recommended strategies from different forecasting agents, leading to a final recommended strategy for users. As different forecasting agents can give rise different prediction results, it is difficult to find a mechanism to handle it. In the literature, the most well known mechanism is majority rule [36] but it can lead to intransitive group preference, which can result in no final recommendation made. We studied how to quantify

the recommendation which is originally in terms of 'Buy', 'Hold' and 'Sell' into newly introduced *prefer ratio*. Using a *score table* to represent different agents' accuracy in different trading period. Also, two weighting methods had been investigated, they are *simple weighting* and *exponential weighting*. Empirical testing shows that the overall system prediction performance can be greatly improved by introducing the Coordinate Agent.

The Technical Agent uses technical analysis to predict the market moves. Technical analysis mainly focuses on analyzing the chart patterns, which is a non-trivial task. Because one time scale alone cannot be applied to all analytical chart pattern discovery processes, the identification of typical patterns on a stock price chart requires considerable knowledge and experience. There have been attempts in the last two decades to solve such non-linear financial forecasting problems using AI technologies such as neural networks, fuzzy logic, genetic algorithms and expert systems which may be accurate, but lack of explanatory power or are dependent on domain experts. This agent is a case based reasoning (CBR) agent that can provide an explainable method of financial forecasting that is not dependent on the input of domain experts. Also, in this agent, we have proposed an algorithm, PXtract which identifies and analyses possible chart patterns, making dynamic use of different time windows. We introduce a wavelet multi-resolution analysis incorporated within a radial basis function neural network (RBFNN) matching method that can be used to automate the chart pattern matching process. The automatic process of identifying stock chart pattern is scarce in literature, our proposed algorithm does well and has achieved an identification rate above 80% on average. We believe that it is helpful to investors.

The Adaptive Agent uses genetic algorithms and artificial neural networks to predict the market moves. Stock price movements are influenced by many factors and indexes, and they may include changes in gold price, prime rates, deposit call, oil price, exchange rates and other factors. Also, every stock has its own characteristics [38]. Building an artificial neural network with fixed input and topology for all stocks is not feasible to obtain accurate prediction results. This agent provides a genetic approach to address the problem. It includes the input selection, network architecture and the output format. A *decision threshold* is also introduced to define the best strategy for decision making by investors. Having introduced the decision threshold, we transformed the forecasting problem into a classification problem and got resolved relatively easier with accurate result.

Finally, with the integrated multi-agent system for different approaches in solving financial market forecasting problems, we believe that the proposed system can benefit both the experienced and novice investors.

Acknowledgements

I would like to express my gratitude to all those who have assisted me and made this thesis possible.

I am deeply indebted to my supervisor, Dr. James N. K. LIU, for his invaluable guidance, helpful advice and constructive criticisms. Without his insight, enthusiasm, encouragement and inspiration, I would not have the courage and determination to complete this work.

Special thanks to two of my schoolmates during my research study, namely Terry Lai and Christine Law, for their considerable help and suggestion. Besides, I would like to thank all the research staffs and students of the department, particularly those who are working with me in the same office, for providing me a comfortable and warm research environment.

Above all, I thank to my dear parents and my brother for their sacrificial love and endless support.

Publications

Raymond W.M. Kwong, James N.K. Liu, *In Proceedings of the CIFEr'03 Workshop, Soft Computing for Financial Analysis: A Genetic Approach Towards Financial Forecasting and Investment Decision Making,* Hong Kong, 2003.

Raymond Kwong, James Liu, *In Proceeding of the 4th ACM Postgraduate Research Day: Chart Patterns Extraction and Recognition using Wavelet and Radial Basis Function Neural Network for Financial Forecasting*, Hong Kong, 2003.

James Liu, Raymond Kwong, Jane You, *In Proceeding of the IASTED International Conference ACI2002: Towards an Intelligent Agents System (iWAF) for e-Finance Application*, Tokyo, Japan, 2002.

James Liu, Raymond Kwong, *In Proceeding of the IASTED International Conference ACI2002: Chart Patterns Extraction and Recognition in CBR System for Financial Forecasting*, Tokyo, Japan, 2002.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Financial data are often represented as a time series of a variety of attributes such as stock prices and different indexes. Time series forecasting is said to be a challenge for many years. Financial market is complex and dynamic, almost every piece of econometric and political information has a certain degree of influence to the market. Continuously taking advantages from the market seems to be non-trivial.

According to the efficient market theory [12] is practically impossible to infer a fixed long-term global forecasting model from historical stock market information. It is said that if the market presents some irregularities, someone will take advantages on it and the irregularities will disappear. But it does not exclude that hidden short-term local conditional regularities may exist; this means that we can still take advantage from the market if we have a system which can identify the hidden underlying short-term irregularities when it occurs.

1.1 Objective

As the rapid advances of technologies and availability of vast information in the Internet, investors are easy to find advices or reference in their financial investment process. Many investment consultants and other financial channels are meant to support different needs of investors, but it is always far from being ideal. Since most investors have not much confidence on these kinds of middle agents because they lack of unbiased market movement analysis, trend prediction, human behaviour and psychological implication, cause and effect studies. Some small investors may even choose just to follow the general trend in making their investment decision. This makes them usually subject to be preies in the volatile market.

The financial market in Hong Kong is a very active market. The average daily turnover of the stock market (Main Board) during 2001 was HK$7.4 billion and HK$5.7 in the first nine months of 2002. According to Hong Kong

Exchanges and Clearing Limited (HKEx), Hong Kong has the world's 9^{th} largest securities market by capitalization and the second largest in Asia after Tokyo. There were 781 listed companies as of May 2002, with a total market capitalization of HK$ 4,010 billion (US$514 billion).

Many ordinary citizens participate in investing money in stocks. Hong Kong Exchanges and Clearing Limited conducted a survey on stock market retail investor participation between November and December 2000 [16]. According to this survey's findings, about 21% of the Hong Kong adult populations (i.e. 1,147,000 individuals) were stock investors. In order to make a profit from the market, investors follow one simple rule: "Buy low, sell high". Although this rule is simple and well known, it is difficult to follow. This is because the trends in the market are influenced by many factors (such as political and economic factors), and on the other hand, the market itself can influence these factors. Various market analysis techniques are applied to interpret the status of the market and predict the market's future trend, but they are not beneficial to small investors because these techniques require a certain degree of expertise in finance and economics. In addition, these techniques require extensive data collection from the market and many calculations, which is too much effort for individual small investors. Therefore, an advisory tool using these techniques is very useful for helping small investors to make trading decisions.

Moreover, the World-Wide-Web provides a means for the users to retrieve information from any Web sites from anywhere in the world, regardless of where the users are located. It can be a basis for uniform information distribution, independent of how information is stored. Information distribution can thus rely on populating knowledge elements on the Web or on a knowledge server on the Web [5]. Unfortunately, our tools for locating, filtering, and analysing that information have not kept pace with the Web's development. E-commerce applications face challenges that intelligent techniques and the adoption of agent technology [30, 23] may overcome. The intelligent techniques include the use of fuzzy logic for knowledge representation and to make useful inferences or actions, expert systems for evidential and heuristics reasoning [29], neural networks for classification and adaptive learning [22, 20], genetic algorithms for solution optimisation [24], and data mining techniques for knowledge discovery [11]. Agent technology allows software modules to be built to monitor, assist, and act on behalf of a user in order to inter-operate with other co-existing agents. It is playing an increasing role in many e-Commerce applications [25].

In order to help to solve this problem, we propose a comprehensive model, an intelligent Web-based agent system (iWAF) to automate a series of processes in support of e-Finance applications. For novice investors, the system supports analysis study of the rationale behind some system recommendations. For experience investors, they can explore various theories for the investment strategies by means of adjusting the weightings, learning parameters, combinations and even some independent variables allocated by different agents in the system, make their own customized advisor. General user can also obtain explanations from the expert agent, therefore formulate their investment strategies upon system recommendations under different financial criteria accordingly.

In addition, we want to study how the emotional behaviour, psychological impacts from the investors will affect market by studying the chart pattern theory in technical analysis [38]. It is said that the emotional and psychological behaviour of the investor can be reflected in the stock chart patterns in technical analysis [38]. This idea is totally absent from the previous studies.

1.2 Background & Literature Review

Some powerful methods are used in solving the non-linear financial forecasting problem, its approach is in at least two directions: One is based on inferring rules from past and current behaviour of market data, it is led by some inductive and learning-based techniques such as neural networks [2] and fuzzy logic [13]. Another direction uses physical and mathematical models [18] based on different economic prototypes. It attempts to find dynamic indicators derived from physical models based on general principles of nonequilibrium stochastic process [31] that reflect certain market factors.

At marco level of forecasting, we need to know the long-term trend of the markets. At the micro level, we need to investigate the market segments and determine the short-term trend and best timing for action. In reality, we need to integrate both and develop both strategies to position the investment. It seems that most of the development in literature tends to focus on part of the forecasting process and has not been able to give much confidence and benefit to investors. More importantly, they always give rise to bias and contradictory results based on somewhat different approaches of analysis. All the literature developments have their own advantages and disadvantages. It is hardly any one of the developments in literature which can give an accurate, understandable and non-bias prediction.

1.3 Methodology

E-finance involves a considerable degree of expertise in financial engineering., network engineering and dynamical systems theory. The awareness of the complex inter-relations between those model parameters and performance metrics is an essential ingredient of successful application development. As a result, a system taking all the advantages of different forecasting approaches is needed.

In this study, a multi-agent framework consisting of five forecasting agents and a coordinate agent is proposed. Different agents will give different recommendations in the investing process, the coordinate agent merges all the recommendations from different agents and provides a final recommendation to the user. It monitors the accuracy of different agents and continues to adjust the weight of the recommendation value of different agents. Agent with higher accuracy will contribute more in the final recommended strategy.

The proposed system consists of five forecasting agents and two non-forecasting agents. Forecasting agents include Fundamental Agent, Technical Agent, Associate Agent, Adaptive Agent and Expert Agent. Different agents give different recommendations in the investment process. Each agent has different mechanism and embedded with different complicated algorithms in it. Completing all the agents in the proposed system is time consuming and need huge efforts to do it. As such, this study just focuses on the construction of the three agents the Coordinate Agent, Technical Agent and Adaptive Agent. These three agents are the most complicated and representative agents in the proposed system.

Chapter 2
Preliminaries

In this chapter, we will review some techniques and technologies which are used in this study.

2.1 Intelligent Agents System

Agents technology has been widely adopted in solving distributed computing problems. Recently, there has a large movement and change of focus in AI research community to apply the AI techniques to distributed computing system. At the very beginning, the focus was just limited to some searching and filtering problems. But as rapid rise on the development, agents now have changed to smart agents which have intelligence and could cooperate each other to achieve a specific task.

In a multi-agent system, each agent autonomously performs different actions. The major difference between an Agent and Object is that Agent has more complicated internal states, and more importantly, it has internal goals. Agents can decide what to do and when to do the assigned tasks. Agents can even reject the tasks assigned but objects have fixed roles to follow. Agents can dynamically change their roles as the application progresses.

2.1.1 Agent Standards

Interoperability between agents is the major concern in developing the agent technology. There are two major agent efforts at standardization standards, they are the Foundation for Intelligent Physical Agents (FIPA) [19] and the Object Management Group (OMG) Agent Working Group [19]. FIPA standard is focused primarily on agent-level issue while OMG is focused on the object-level interoperability and management.

2.1.2 FIPA

The FIPA standard governs three areas, they are agent management, agent communications and agent software integration. The FIPA agent management reference model defines only basic infrastructure. An agent platform provides an Agent Management System (AMS) that controls the agent life-cycle, a Directory Facilitator (DF) provides yellow-pages lookup services, and a Message Transport System (MTS) provides internal agent to agent communications as well as external messaging with other FIPA-compliant agent platforms [19].

2.1.3 OMG

The Object Management Group leads the area of mobile agent system. The Mobile Agent System Interoperability Facilities (MASIF) specification approved in 1998 defines how agents can migrate between agent platforms using CORBA Interface Definition Language (IDL) interfaces and the CORBA security, naming and life cycle services.

2.2 Neural Network

Artificial neural networks (ANNs) is one of the soft computing technologies that proved to be promising in solving most of pattern recognition and classification problems. It exhibits a number of desirable properties not found in conventional symbolic computation system including robust performance when dealing with noisy, incomplete and general input patterns. It mimics the human brain's ability to recognize pattern, make predictions and decisions based on past experience.

ANNs are said to be the model of human brain, the basic unit of it is called Neurons. Typically, a ANN neuron or computing element is basically a comparator that produces an output when the cumulative effect of the input stimuli exceeds a threshold value. A single neuron with three inputs and one output is illustrated in Figure 2.1

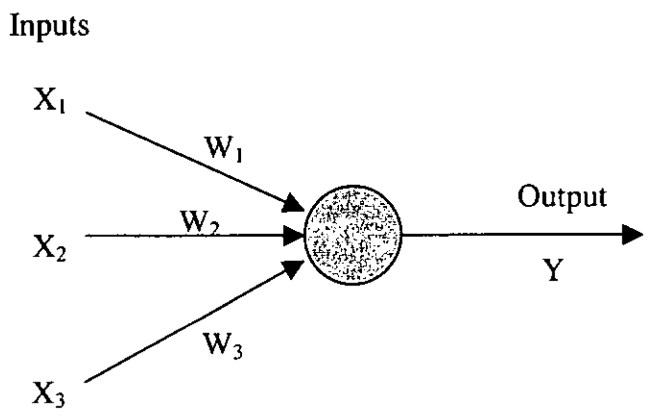


Figure 2.1 - A Simple Artificial Neural Network

The output of the artificial neuron will be processed by an activation function before passing it to the other neuron. Figure 2.2 illustrates the details

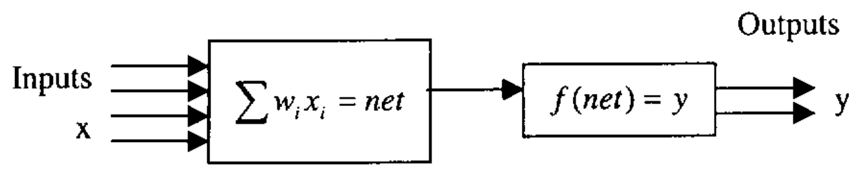


Figure 2.2 - Schematic Diagram of a Single Neuron

Activation function is a rule for calculating an output value that will be transmitted to other neurons or for presenting to the environment. The output is referred to as the activation for the neuron. The activation may be a real number that is restricted to some intervals such as 0 to 1 or some discrete number such as {0,1} or {+1, -1}. The output range depends on what activation function is used and should be problem dependent.

Binary sigmoid function is one of the most typical activation functions whose output is in the range 0 to 1. Figure 2.3 illustrates the graphical representation of Binary sigmoid function.

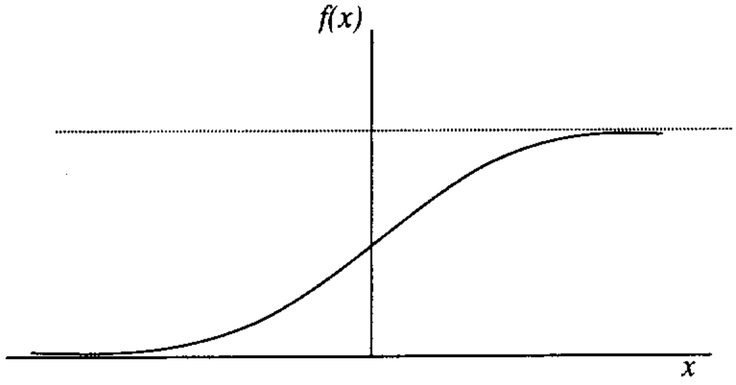


Figure 2.3 - Binary Sigmoid Activation Function

$$f(x) = \frac{1}{1+\exp(-x)}$$

$$f'(x) = f(x)[1 - f(x)]$$

2.2.1 Multi-Layer Feed-Forward Back-propagation Neural Network

Multi-Layer Feed-Forward Neural Network has more than one layer of neurons, typically consisting of three layers. They are called *input layer*, *hidden layer* and *output layer*. Each neuron in the *input layer* is connected to the neurons in the *hidden layer*. In some networks, the number of *hidden layers* can be occasionally more than one, it depends on the network architecture design. Each *hidden layer* is connected to each other and the last one is connected to the *output layer*.

The architecture of the network defines how to connect the neurons in the network. The nodes are organized into a series of layers with an *input layer*, one or more than one *hidden layers*, and an *output layer*. In the network, data follow in one direction only, from the *input layer* to the *output layer*. Figure 2.4 illustrates the typical three layers feed-forward neural network.

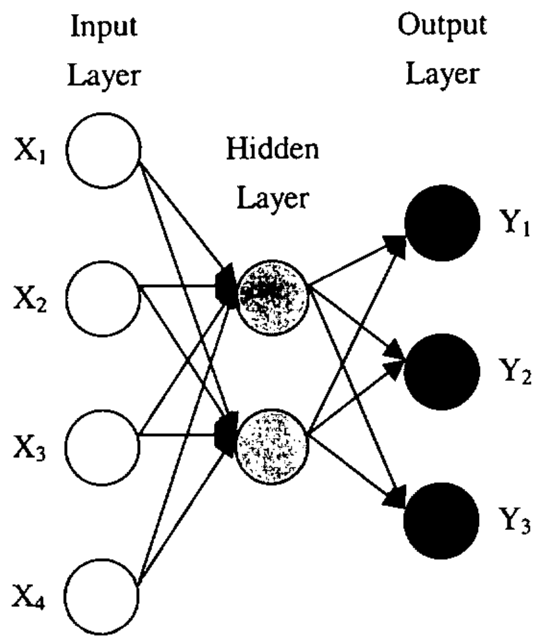


Figure 2.4 - Typical Three Layers Feed-forward Neural Network

Training is one of the most important procedures in building the network model. The accuracy of the network depends on the data input, network architecture and the training algorithm. One of the most common algorithms is *backpropagation* (BP). BP is a supervised, greedy training algorithm, its target on decreasing the total errors as the number of epochs increases. With each epoch, the weights are modified to decrease the error on the training patterns. As training progresses, the amount of change in the error function becomes smaller. BP method can be applied to any multiplayer network that uses differentiable activation functions. It amounts to repeatedly adjusting the interior layer weights using the computed errors from the *output layer* and propagating the error adjustments backwards layer-by-layer to the *input layer.* Convergence is not assured for BP and it may get stuck to a local minimum. To solve this problem, a momentum [6] is introduced to prevent it from being trapped by a local minimum and increase the training efficiency.

2.2.2 Radial Basis Function Neural Networks (RBFNNs)

Radial Basis Function Neural Networks (RBFNNs) is a kind of feed-forward neural networks. It has similar capabilities of traditional feed-forward neural networks, and the ability to solve any function approximation problem. Park and Sandberg [34, 33] proved that RBFNNs have the capabilities of universal approximation. Under certain mild conditions on the RBFs, RBFNNs are capable of approximating arbitrarily well any function.

RBFNNs consist of three layers of neurons, they are *input layer, hidden* *layer* and *output layer*. RBFNNs have radial symmetry, each *hidden node* in the *hidden layer* is associated with a radial symmetry function (a very popular choice is the Gaussian Functions), with appropriate centers (means) and the auto-covariance matrices. Figure 2.5 illustrates the architecture of a typical RBFNN. Function $g_i(x)$ in the hidden layer is the radial symmetry function. Each function in different nodes can have a different center (mean). It is noted that there has a bias node (node 0) in the hidden layer designated as $g_0(x)$ for uniformity of notation; $g_0(x)$ is equal to 1, and node 0 in the hidden layer is not connected to any node in the input layer.

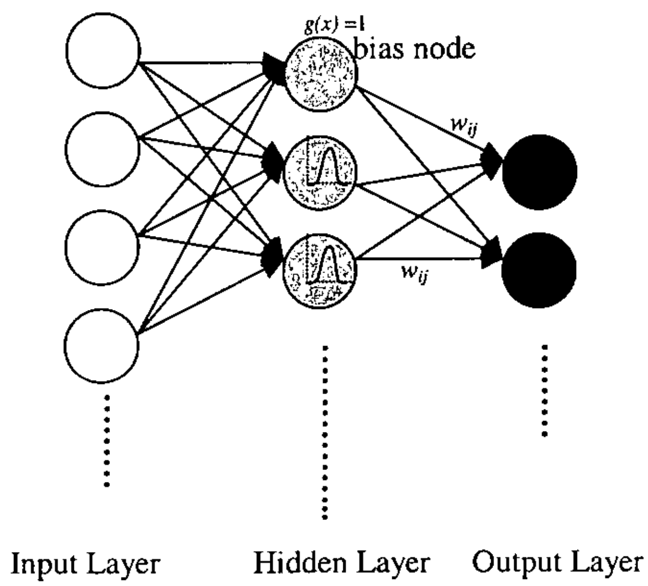


Figure 2.5 - A Typical Example of RBFNN

Training in RBFNNs is not only done by continuously adjusting the connection weights but also the center (mean) and the auto-covariance matrices in the hidden layer. Initially, a center (mean) in the radial symmetry function can be chosen at random from the input data. For each epoch, it can be updated by using K-means Clustering methods [4], Self-Organizing Feature Map Clustering Procedure [4], or some supervised gradient descent procedure. Here's the equation for training the network by using supervised gradient descent procedure.

$$\Delta w_{ij} = \eta_w [d_i(p) - y_i(p)] g_j [x(p)]$$

$$\Delta v_j = \eta_v \sum_{l=1}^{l} [d_l(p) - y_l(p)] w_{lj} g_j [x(p)] \sum_j^{-1} (x(p) - v_j)$$

$$\Delta \Sigma_j^{-1} = \eta_\Sigma \sum_{l=1}^{l} [d_l(p) - y_l(p)] w_{lj} g_j [x(p)] (-1)(x(p) - v_j)(x(p) - v_j)^T$$

where η_w, η_v and η_Σ are the corresponding learning rates of the parameters w (weights), v (center/mean), and Σ (variance), respectively. Functions $d_l(p)$ and $y_l(p)$ are the network output and the desired output for the training pattern p respectively.

2.3 Wavelet Analysis

Wavelet analysis has been widely applied in the areas of signal processing, image processing and pattern recognition with encouraging results [44]. It was the development of applied mathematics in 1980s introduced by J.Morlet. It denotes a univariate function ψ, defined on $\Re$ when subject to fundamental operations of shifts and dyadic dilation, yielding an orthogonal basis of $L^2(\Re)$.

The orthonormal basis of compactly supported wavelets of $L^2(\Re)$ is formed by the dilation and translation of a single function ψ(x).

$$\psi_{j,k}(x) = 2^{\frac{-j}{2}} \psi(2^{-j} x - k)$$

where j, $k \in Z$. Vanishing moments denote that the basis functions are chosen to be orthogonal to the low degree polynomials. It is said that a function $\psi(x)$ has a vanishing k^{th} moment at point t_0 if the following equality holds with the integral converging absolutely:

$$\int (t - t_0)^k \psi(t) dt = 0$$

The function $\psi(x)$ has a companion, the scaling function $\phi(x)$, and these functions satisfy the following relations:

$$\phi(x) = \sqrt{2}\sum_{k=0}^{L-1} h_k \phi(2x-k),$$

$$\psi(x) = \sqrt{2}\sum_{k=0}^{L-1} g_k \phi(2x-k),$$

where h_k and g_k are the low-pass filter and high-pass filter coefficients respectively, L is related to the number of vanishing moments k and L is always even. For example, $L = 2k$ in the Daubechies wavelets.

$$g_k = (-1)^k h_{L-k-1}, \quad k = 0, \ldots, L-1$$

$$\int_{-\infty}^{+\infty} \phi(x)dx = 1,$$

The filter coefficients are assumed to satisfy the orthogonality relations:

$$\sum_n h_n h_{n+2j} = \delta(j),$$

$$\sum_n h_n g_{n+2j} = 0,$$

for all j, where $\delta(0) = 1$ and $\delta(j) = 0$ for $j \neq 0$

2.3.1 Multi-resolution Analysis

Multi-resolution analysis (MRA) was first published in 1989 by Mallat and Meyer [32]. It was formulated based on the study of orthonormal, compactly supported wavelet bases.

The wavelet basis induces a MRA on $L^2(R)$, the decomposition of the Hilbert space $L^2(R)$, into a chain of closed subspaces V_i.

$$\cdots \subset V_4 \subset V_3 \subset V_2 \subset V_1 \subset V_0 \subset \cdots$$

such that

- $\bigcap_{j\in Z} V_j = \{0\}$ and $\bigcup_{j\in Z} V_j = \mathrm{L}^2(R)$

- $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$

- $f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0$

- $\exists \psi \in V_0, \{\psi(x-k)\}_{k\in Z}$ is an orthogonal basis of V_0

In pattern recognition, an 1-D pattern, $f(x)$, can always be viewed as a signal of finite energy; such that,

$$\int_{-\infty}^{+\infty} |f(x)|^2 < +\infty$$

It is mathematically equivalent to $f(x) \in L^2(R)$. It means that MRA can be applied to the function $f(x)$ and decompose it to $L^2(R)$ space. In MRA, closed sub-space V_{j-1} can be decomposed orthogonally as:

$$V_j = V_{j+1} \oplus W_{j+1} \qquad (2.1)$$

V_j contains the low-frequency signal component of V_{j-1} and W_j contains the high frequency signal component of V_{j-1}. According to the wavelet orthonormal decomposition as shown in equation 2.1, first V_j is decomposed orthogonally into a high-frequency sub-space V_{j+1} and W_{j+1}. The low-frequency sub-space V_{j+1} is further decomposed into V_{j+2} and W_{j+2} and the processes can be continued. The above wavelet orthonormal decomposition can be represented by

$$\begin{aligned} V_j &= W_{j+1} \oplus V_{j+1} \\ &= W_{j+1} \oplus W_{j+2} \oplus V_{j+2} \\ &= W_{j+1} \oplus W_{j+2} \oplus W_{j+3} \oplus V_{j+3} \\ &= \ldots\ldots \end{aligned}$$

Projective operators A_j and D_j are defined as:

$A_j : L^2(R) \Rightarrow V_j$ projective operator from $L^2(R)$ to V_j

$D_j : L^2(R) \Rightarrow W_j$ projective operator from $L^2(R)$ to W_j

Since $f(x) \in V_j \subset L^2(R)$,

$$f(x) = A_j f(x) = \sum_{k \in ZZ} c_{j,k} \phi_{j,k}(x)$$

$$= A_{j+1} f(x) + D_{j+1} f(x)$$

$$= \sum_{m \in ZZ} c_{j+1,m} \phi_{j+1,m}(x)$$

$$+ \sum_{m \in ZZ} d_{j+1,m} \psi_{j+1,m}(x)$$

2.3.2 The Fast Wavelet Transform

Daubechies [8] has discovered that the wavelet transform can be implemented with a specially designed pair of Finite Impulse Response(FIR) filters called a "Quadrature Mirror Filter" (QMF) pair. The output of the QMF filter pair is down sampled by a factor of two, the low frequency filter output is fed into another identical QMF filter pair. This operation can be repeated as a pyramid algorithm, yielding a group of signals that divides the spectrum of the original signal into octave bands with successively coarser measurements in time as the width of each spectral band narrows and decreases in frequency.

$$c_{j+1,m} = \sum h_k c_{j,k+2m} \qquad (2.2)$$

$$d_{j+1,m} = \sum g_k c_{j,k+2m} \qquad (2.3)$$

2.3.3 Wavelet Families

There are many different kinds of wavelets, it can mainly be distinguished into five types [17], they are:

a) Orthogonal wavelets with FIR filters

   These wavelets can be defined through the scaling filter. Predefined families of such wavelets include: *Haar, Daubechies, Coiflets,* and *Symlets.*

b) Biorthogonal wavelets with FIR filters

   These wavelets can be defined through the two scaling filters *wr* and *wd*, for reconstruction and decomposition respectively. The *BiorSplines* wavelet family is a predefined family of this type.

c) Orthogonal wavelets without FIR filter, but with scale function

   These wavelets can be defined through the definition of the wavelet function and the scaling function. The *Meyer* wavelet family is a predefined family of this type.

d) Wavelets without FIR filter and without scale function

   These wavelets can be defined through the definition of the wavelet function. Predefined families of such wavelets include: *Morlet* and *Mexican hat.*

e) Complex wavelets without FIR filter and scale function

   These wavelets can be defined through the definition of the wavelet function.

Predefined families of such wavelets include: *Complex Gaussian* and *Shannon*.

In this study, we put the focus on the first type of wavelet (Orthogonal wavelets with FIR filters) as it has the capability to solve one dimensional signal processing problems and with easier implementation. The four orthogonal, with FIR filters wavelets are illustrated in Figure 2.6.

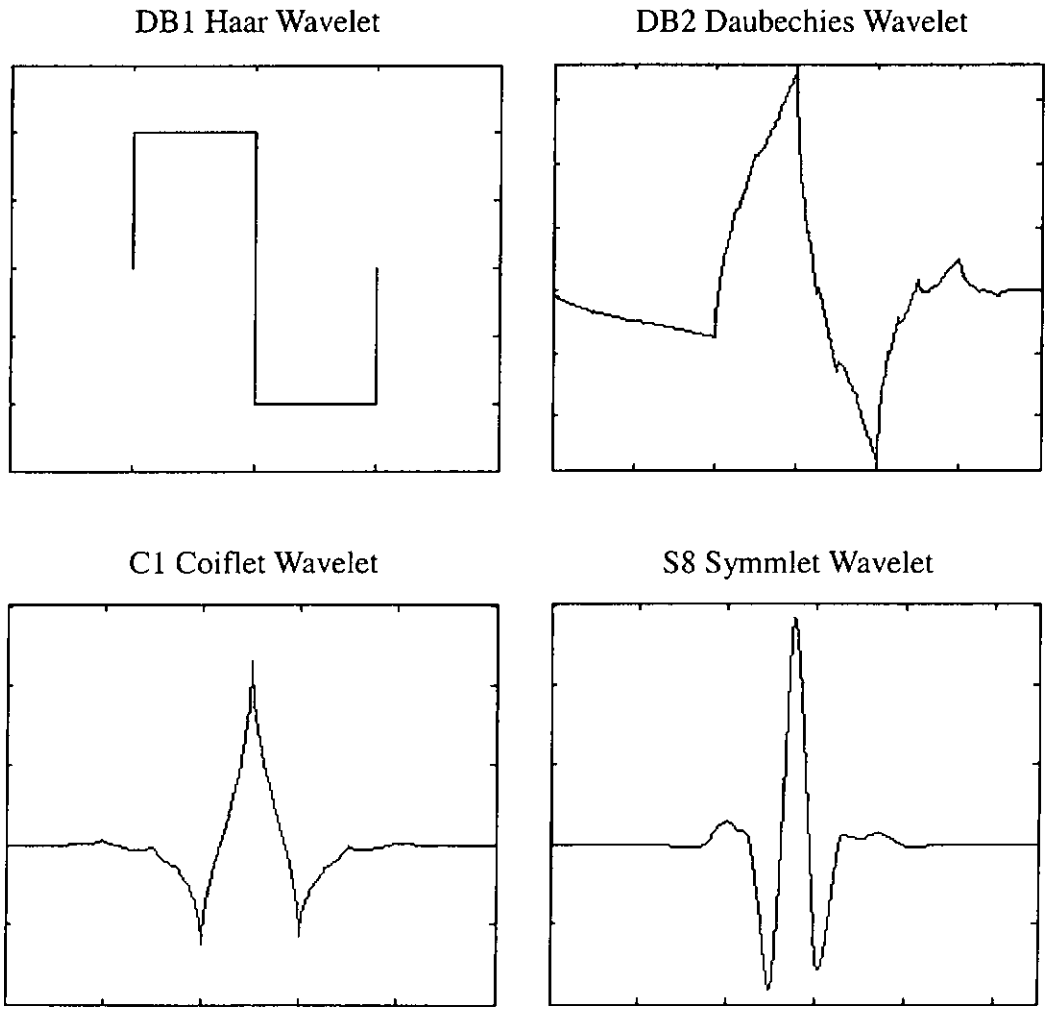


Figure 2.6 - Wavelet Families Used in This Study

2.4 CBR System

Case Based Reasoning (CBR) is a useful form of problem solving because it can reuse previous problem solving in new circumstances [40]. Its approach is attractive because human problem solving is often case-based and so the user can recognize the justification it uses to support its solutions. A case based reasoning system usually consists of four phases, they are case identification, case retrieval/adaptation, reasoning and learning. In the case identification phases, the most frequently used model is features-based [40]. A feature vector is usually being used to represent a case. In the case retrieval and adaptation, there have many different kinds of similarity measure for case matching. The most frequently used method is Euclidean distances. After a case has been analyzed, the results will be stored and used as a referring case in the next case adaptation process.

Chapter 3
Development of iWAF

The suggested system consists of five forecasting agents and two non-forecasting agents. Forecasting agents include Fundamental Agent, Technical Agent, Associate Agent, Adaptive Agent and Expert agent. Non-forecasting agents include Information Gathering Agent and Coordinate Agent. IBM Aglets [7] act as the agent communication 'backbone', Java Servlets is used as the communication channel between the agents and the users. An overview of the system is shown in Figure 3.1.

Different agents pays different roles in the system, they are:

1. Fundamental agent performs fundamental analysis using information such as P/E ratio, book values in the case of stocks, crop reports or import/export

figures in the case of commodity futures, a company's finance, forecasts of sales, earnings, and expansion plans.

2. Technical agent applies numerical techniques such as moving averages, curve fitting, RSI, OBV, complex stochastic models, etc to try to derive the market movement.

3. Adaptive agent synthesizes those economic factors and pattern identification.

4. Associate agent investigates the dynamics of human factors and psychological impacts.

5. Expert agent which processes AI heuristics for the model of trading rules, risk and return calculation, and investment strategies acting on indicators derived from the market. It also provides personalized advice and substantiation of recommendations.

6. Information Gathering agent (IGA) and Information Filtering agent (IFA) track and filter financial information for the analysis and system training, it also provides all the information that the forecasting agents need.

7. Coordinate agent to collect all the recommended strategies from different forecasting agents, leading to a final recommended strategy for user.

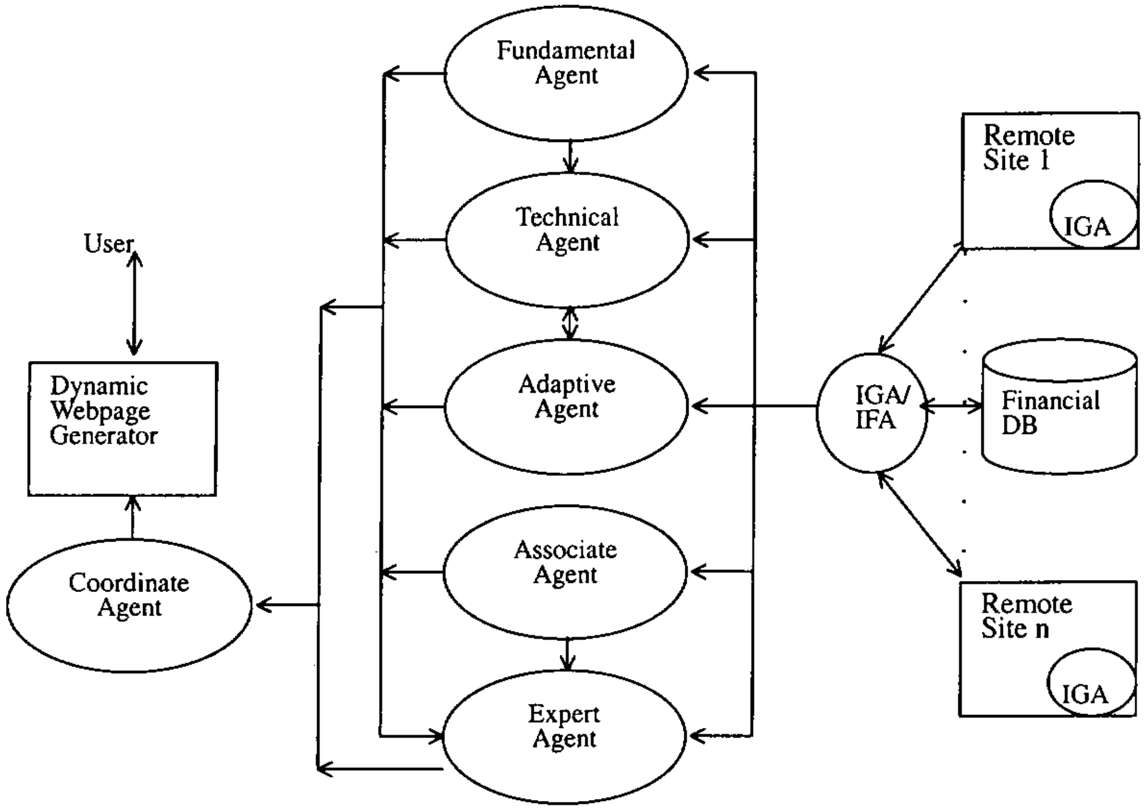


Figure 3.1 – A Conceptual Framework for iWAF

3.1 Forecasting Agents

Each forecasting agent gives its advice on the possible strategies: buy, hold and sell according to its analysis result.

3.1.1 Fundamental Agent

Fundament analysis is to determine the market direction in which to trade and technical analysis is to keep track of the time for entry and exit of such trades. It lets us know the general market direction, mainly making use of some popular financial ratios [1] that in liquidity, leverage, profitability, and turnover dimensions. The agent works with Technical agent and Expert agent.

3.1.2 Technical Agent

Technical analysis is a way of studying price behavior. It uses quantifiable information in terms of carts to predict trends [35]. Technical analysis uses some important measures of how a stock is acting in relation to the overall market and they include relative strength (price of stock / price of market average), on balance volume (OBV), stock chart patterns [38] and other stochastic indicators.

All these indicators can be easily obtained but not the stock chart patterns which are experience demanding and non-trivial to discover the patterns in the time series data. Successful identifying the patterns is one of the key components in technical analysis which plays a very important role [38] to reveal the market trends.

Different chart patterns are said to reveal different market trends. For example, a Head-and-Shoulders Tops chart pattern is said to reveal that the market will most likely to have a 20% to 30% rise in the coming future [38]. Identifying this chart pattern is crucial in positioning the investment move.

Technical agent synthesizes all these financial indicators with the chart pattern to forecast the market trends.

3.1.3 Adaptive Agent

The construct of the adaptive agent is similar to that in Liu and Sin [28]. The model incorporates fuzzy logic and an expert system for data preprocessing. Statistical techniques and genetic algorithm are employed for data transformation, variable selection, and model dimension reduction [27]. Artificial neural networks are used for knowledge extraction and generalization. Fuzzy logic is then applied to the system to derive the forecast. The adaptive agent will investigate the influences of some economic factors (including inflation rate, interest rate, unemployment rate and consumer index) to the market movements.

3.1.4 Associate Agent

Human factors and psychological impacts reflect some investors' action in the stock market. For example, it is a statistical fact that stocks decline faster than they rise. Fear causes a panic reaction while greed takes a while to simmer. The CBR agent model [26] has been considered to decode stock signals as unit waves according to Elliott wave theory, and the agent is set to refer to successive cases within a series of stock cases as one complete sample for prediction purpose. Elliot theory explains about the impulse waves, corrective waves which are combined to form a middle to long-term corrective waves. The cases reflect some degree of reaction of investors in the market, representing some human factors and psychological impacts somehow.

3.1.5 Expert Agent

Expert Agent collaborates analysis results from all the other forecasting agents and formulates appropriate investment strategies. Some investment heuristics [41] for consideration include: using buy-stop-limit order to buy stock when it breaks out within a limit above a certain level; not to buy a stock when good news comes out, especially if the analysis shows a significant advance prior to the news release; not to buy a stock because it appears cheap after getting smashed; not to buy a stock in a downtrend; not to hold a stock that is in a downtrend no matter how low is the price/earning ratio; never trust a breakout that isn't accompanied by a significant increase in volume; never buy a stock, no matter how good are the other factors, if the relative strength is in negative territory and it remains in poor shape; never buy a stock that has heavy nearby overhead resistance.

3.2 Non-forecasting Agent

Non-forecasting agents include the Coordinate Agent and Information Gathering Agent. The intention of having these two agents is to provide assistance for the forecasting agents, in a form of information provider, synchronizer and decision maker.

3.2.1 Coordinate Agent

Coordinate agent collects all the recommended strategies from different forecasting agents, leading to a final recommended strategy for user. Details are described in Section 4.

3.2.2 Information Agent

Information Gathering agent tracks and filters financial information for the analysis and system training, it also provides all the information that the forecasting agents need.

3.3 System Overview

In this study, the framework of the intelligent multi-agents system (iWAF) has been developed. The system consists of five forecasting agents, an information gathering agent and a coordinate agent. Forecasting agents including fundamental, technical, adaptive, associate and Expert agent. Different agents give different recommendations in the investment process.

Furthermore, completing all the agents in the proposed system is time consuming and need huge efforts to do it. As such, this study just focuses on the construction of the three agents the Coordinate Agent, Technical Agent and Adaptive Agent. These three agents are the most complicated and representative agents in the proposed system.

Chapter 4
Coordinate Agent

As different forecasting agents can give rise different prediction results, the coordinate agent introduces a mechanism to collect all the prediction results and provide a final recommendation for users. We measure each forecasting agent's accuracy based on the historical performances. Coordinate agent makes reference of the prediction accuracy of each agent to form score tables. Typical score tables are shown in Tables 4.1 and 4.2.

Agent	Score/Accuracy
Fundamental	50%
Technical	70%
Adaptive	80%
Associate	60%
Expert	80%

Table 4.1 – Sample Scores by Different Agents During General Period

Agent	Score/Accuracy
Fundamental	60%
Technical	40%
Adaptive	40%
Associate	80%
Expert	80%

Table 4.2 – Sample Scores by Different Agents During War Period

We can have one or more score tables to represent different time periods. As we note that the stock market may show strange movement in some period and this can be caused by some emotional behaviour or psychological effect from the investors. The most obvious case is during the war crisis or political instabilities of the government. Stock values in this period will have dramatic fluctuation; it may have sudden drop or rise. Usually the traditional fundamental or technical analyses may fail in this period. Different score tables are used for representing the accuracy of different agents in different periods. To maintain the score tables, the coordinate agent gathers all the prediction results of each agent and automatically updates the score tables when necessary.

Each forecasting agent will advice one of the three investment strategies: Buy, Hold and Sell. Consider a case with three agents and it holds the following strict preferences.

Agent a: Buy > Sell > Hold

Agent b: Sell > Hold > Buy

Agent c: Hold > Buy > Sell

If we use the simple majority rule [36] to make decision, it will lead to the following:

Buy > Sell, since 2 out of 3 agents prefer Buy to Sell.

Sell > Hold, since 2 out of 3 agents prefer Sell to Hold.

Hold > Buy, since 2 out of 3 agents prefer Hold to Buy.

So using the simple majority rule can lead to intransitive group preference. Thus, we introduce a *prefer ratio* for the advice from each agent. Prefer ratio P_{iB}, P_{iH} and P_{iS} that represent how much percentage agent i suggests to Buy, Hold and Sell respectively. After then, the advice given out from each forecasting agent will become the percentage values rather than just a simple preference.

The weight of each agent in the final result is based on the score table, the score of agent i is denoted by S_i. Two approaches of calculating the weight of each agent are considered. One is based on the *simple weighting* method as shown in equation 4.1. The other is considered that the dependency of a more accurate agent is much more than the others (accuracy is obtained by continuous assessment). The change of dependency is considered has an exponentially increasing, the *exponential weighting* method is shown in equation 4.2.

Differences between the two methods are discussed later in Section 4.3.

$$W_i = \frac{S_i}{\sum_i S_i} \tag{4.1}$$

$$W_i = \frac{e^{S_i}}{\sum_i e^{S_i}} \tag{4.2}$$

For each forecasting agent, three prefer ratios of Buy, Hold and Sell is calculated respectively. We have P_{iB}, P_{iH} and P_{iS} ($0 \leqq P_{iB}, P_{iH}, P_{iS} \leqq 1$ & $P_{iB} + P_{iH} + P_{iS} = 1$). The overall recommendation is based on the weighting and the three prefer ratios of each agent. The final prediction value of agent i for each strategy is computed by equation 4.3.

Among the three final scores for Buy(S_{buy}), Sell(S_{sell}) and Hold(S_{hold}), we simply choose the one with the highest score as the final recommendation. Since the one with higher accuracy will increase the weight of the corresponding agent, the final recommendation value depends much more on that particular agent.

$$\begin{aligned} S_{buy} &= \sum_i^N W_i \cdot P_{iB} \\ S_{sell} &= \sum_i^N W_i \cdot P_{iS} \\ S_{Hold} &= \sum_i^N W_i \cdot P_{iH} \end{aligned} \tag{4.3}$$

4.1 Experiments and Evaluation

Simulations have been carried out to find the impact of the coordinate agent. We set up the simulation with initial stock value of 100. Maximum daily variation of the stock price is set to be 5%. Before the simulations start, we defined a term 'best strategy' which means that the user will have the greatest profits by following the action suggested. If the price increases a half of the maximum variation (2.5%), we assume that the best strategy is said to be Buy, if the price decreases a half of the maximum variation (-2.5%), assume that the best strategy is Sell, otherwise, Hold. We assigned different accuracy to different agents which give different strategies. Coordinate agent collects all the results and provides a final recommendation by using equation 4.3 in section 4. The final recommendation is then compared with the best strategy mentioned earlier. If matched, it is considered as a correct prediction. The stock performance is being simulated over 10000 trading days and the results are shown in Tables 4.3, 4.4 and 4.5.

The accuracy of the recommendation is measured as the percentage of the coordinate agent's strategy that matches the best strategy. It is found that the coordinate agent results lower accuracy when each agent has an accuracy lower than 50%. It is more obvious when the agents have poor accuracy (20%). Conversely, when each agent has accuracy higher than 50%, the accuracy will be increased, typically by a 10% increase when each agent in the system has relatively high accuracy (70%, 80% and 90% respectively).

However, according to Table 4.3, if there has one particular agent which has an outstanding accuracy, coordinate agent results with a lower overall accuracy. Agents with relatively lower accuracy are considered has noises in this case. It is found that if the difference in accuracy between the agents is more than 20%, the results from the agents with relatively lower accuracy should be discarded in order to eliminate the noises. Another method to eliminate the noises is to use the exponential weighting as shown in equation 4.2 in Section 4. In a decision making process, it is said that it is natural to rely much more on the one which is more trustable. Exponential weighting has this characteristic. It is found that using exponential weighting can eliminate the noises problem mentioned earlier. Differences between using simple weighting and exponential weighting are shown in Table 4.4.

Table 4.5 shows the impact when there has only 3 forecasting agents in the system. It is found that it is more accurate than using 5 forecasting agents if each agent has accuracy lower than 50%. But when the agents have accuracy higher than 50%, it improves the accuracy. It is found that the amount of improvement using three agents is smaller than that using five agents.

Agent	Accuracy (%)													
A	20	40	50	60	70	80	90	60	70	70	80	80	90	90
B	20	40	50	60	70	80	90	50	50	60	60	70	70	80
C	20	40	50	60	70	80	90	50	50	60	60	70	70	80
D	20	40	50	60	70	80	90	50	50	60	60	70	70	80
E	20	40	50	60	70	80	90	50	50	60	60	70	70	80
Coordinate	7	32	48	**66**	**80**	**91**	**98**	**53**	**57**	**69**	**73**	83	**86**	94

Table 4.3 – Simulation Results Using Five Forecasting Agents
(Using Simple Weighting)

Agent	Accuracy (%)													
A	20	40	50	60	70	80	90	60	70	70	80	80	90	90
B	20	40	50	60	70	80	90	50	50	60	60	70	70	80
C	20	40	50	60	70	80	90	50	50	60	60	70	70	80
D	20	40	50	60	70	80	90	50	50	60	60	70	70	80
E	20	40	50	60	70	80	90	50	50	60	60	70	70	80
Coordinate	7	32	48	**66**	**80**	**91**	**98**	**57**	**69**	**71**	**80**	84	**90**	94

Table 4.4 – Simulation Results Using Five Forecasting Agents
(Using Exponential Weighting)

Agent	Accuracy (%)													
A	20	40	50	60	70	80	90	60	70	70	80	80	90	90
B	20	40	50	60	70	80	90	50	50	60	60	70	70	80
C	20	40	50	60	70	80	90	50	50	60	60	70	70	80
Coordinate	11	35	49	**62**	**75**	**86**	**95**	57	69	**69**	**78**	**80**	**89**	**90**

Table 4.5 – Test results Using Three Forecasting Agents (Using Exponential Weighting)

Simulations have been carried out to find the impact of the coordinate agent. The result of the experiments shows that the coordinate agent has an improvement of more than 10% of accuracy in a situation when each forecasting agent has accuracy higher than 70%.

4.2 Experiments on real financial data

In this section, we carried out experiment with 5 stocks with 2 agents: technical and adaptive agents. The technical and adaptive agents are given in the section 5 and 6. Before the experiments carried out, we find out the two agents' accuracy with 5 different stocks in 6 years period (1 Jan 1990 – 31 Dec 1995), period 1 Aug 1990 – 31 April 1991 is categorized as war period (Gulf war). Table 4.6 shows the 5 different stocks in used.

Stock ID	Stock Name
00341	CAFÉ DE CORAL HOLDINGS LTD.
00293	CATHAY PACIFIC AIRWAYS LTD.
00011	HANG SENG BANK LTD.
00005	HSBC HOLDINGS PLC.
00016	SUN HUNG KAI PROPERTIES LTD.

Table 4.6 – The Five Different Stocks and its Stocks ID

Scores by Different Agents During General Period
(1 Jan 90 – 31 July 90 &
1 May 91 – 31 Dec 95)

Scores by Different Agents During War Period
(1 Aug 90 – 31 Apr 91)

Table 4.7 – Score Table of the Technical and Adaptive Agent over the Period of 1 Jan 1990 – 31 Dec 1995

Stock ID	With Coordinate Agent	Technical Agent Only	Adaptive Agent Only
00341	63.9%	64.2%	60.2%
00293	64.5%	61.6%	63.4%
00011	66.6%	67.3%	62.1%
00005	66.7%	64.8%	63.3%
00016	63.8%	62.7%	59.4%

Table 4.8 - Testing Results on the Trading Days of 1 Nov 2000 – 31 Oct 2001

In order to test the impact of the Coordinate Agent, we have carried out three tests as shown in Table 4.8. We found that the system with coordinate agent is generally the one obtaining a more accurate result.

Chapter 5
Technical Agent

Technical analysis mainly focuses on analyzing the chart patterns, which is a non-trivial task. Because one time scale alone cannot be applied to all analytical processes [2], the identification of typical patterns on a stock price chart requires considerable knowledge and experience.

Chart patterns identification is one of the key elements in technical analysis. In literature, Chung [14] conducted extensive study in segmentation of time series data. Instead of using a fixed length approach, the segmentation uses a dynamic approach and that is genetic algorithm. Each segment in the time series data is extracted according to the predefined pattern templates. The identified segment is then matched with the predefined pattern templates using the Perceptually Important Points (PIPs) methods [15]. However, the above technique strongly depends on some parameters: desired segment length *dlen* and desired length control *dlc*; which have no optimal values and will work for all time series data. Another study uses a recurrent neural network [21] which focuses on identifying only the triangle patterns [38] in the stock data. Although the identification rate is encouraging, it has no extensibility to identify others chart patterns.

Research on the automatic process of identifying stock chart patterns is even scarcer. As far as we know, there is no research reported in the literature that explicitly deals with our problem. Our proposed system is a case-based reasoning system, it can be effective even if the knowledge base is imperfect. Certain techniques of automated learning, such as explanation-based learning, work well only if a strong domain theory exists. In contrast, case reasoning can use many examples to overcome the gaps in a weak domain theory while still taking advantage of domain theory. CBR can also be used when the descriptions of the cases, as well as the domain theory, are incomplete. A further advantage of CBR is the relative ease of combining techniques with other approaches such as production rules. An example of such compatibility is a system that whenever possible uses case reasoning to solve problems.

Novice investors usually fail to identify the pattern correctly, this makes them fall into prey in the stock markets. In this agent, CBR with a new algorithm PXtract is proposed to solve the problem. Pxtract, which identifies and analyses possible chart patterns, making dynamic use of different time windows, and applied wavelet multi-resolution analysis incorporated within a radial basis function neural network (RBFNN) matching method that can be used to automate the chart pattern identifying process. The next section shows the background information of the technical indicators, which is used in this technical agent and Sections 5.2 and 5.3 show the details of the proposed CBR methods and the details of the brand new algorithm PXtract.

5.1 Technical Indicators

Fundament and technical analysis are two important tools in financial market prediction. Analysts use fundamental analysis to forecast prices based on economic data such as P/E ratio, book values, crop reports and import/export figures [41]. Technical analysis is a way of studying price behavior. It uses quantifiable information in terms of carts to predict trends [35]. Other important measures of how a stock is acting in relation to the overall market include relative strength (price of stock / price of market average), on balance volume (OBV) and other stochastic indicators. The integration of both analytical techniques: fundamental analysis and technical analysis often gives a better prediction result [37].

The following theories are considered as part of the technical analysis used in this study.

5.1.1 Moving Average (MA)

Moving average represents a smoothing of actual price fluctuations, done by averaging the last L closing prices. It smoothes out a data series and makes it easier to identify the direction of a trend. It has little predictive value. There are types of moving average, they are simple and exponential moving average. The formula for a simple moving average is as follows:

$$\text{Simple Moving Average} = \frac{\text{Sum of } L \text{ day's Closing Price}}{L}$$

The simple moving average obviously has a lag, but some investors prefer simple moving averages over long time periods to identify changes of long-term trend. In contrast, exponential moving averages may be prone to quicker breaks. Some traders prefer to use exponential moving averages for shorter time periods to capture changes quicker.

$$\text{Exponentia l Moving Average} = \sum_{K=0}^{L-1} \frac{\text{Closing Price}}{\dfrac{\beta - \beta^K}{1-\beta}} \qquad \beta \in [1,0]$$

Smoothing gives a heavier weight to the most recent time periods and a lesser weight to earlier ones. If $\beta = 1.0$, then the average is a simple moving average.

5.1.2 Relative Strength Indicator (RSI)

The relative strength index (RSI) measures the strength of a stock movement. The concept of momentum provides the theoretical basis for Relative Strength Index. The index uses a momentum oscillator to measure the velocity or rate of change of price over time.

To construct a relative strength index, a time window k should be chosen.

$$RS = \frac{\text{Average of } k \text{ days close UP}}{\text{Average of } k \text{ days close DOWN}}$$

$$RSI = 100\left(1 - \frac{1}{1 + RS}\right)$$

RSI oscillates from 0 to 100. Usually, the value of RSI above 70 is regarded as overbought and below 30 as oversold and may be used as an early warming signal.

5.1.3 On Balance Volume (OBV)

A popular indicator, OBV is a running total of transaction volume that reflects accumulation or distribution. The calculation is done by keeping a running total of the daily volume. If the market closes up on the day, we add the daily volume a cumulative total, and if the market closes down on the day, we subtract the daily volume. The direction of the OBV indicator is used to analyze the trends. If the direction trends show increasing, it is believed that prices will increase. If decreasing, it is believed that prices will fall.

5.1.4 Chart Patterns

According to Thomas [38], there are up to 47 different chart patterns which can be identified in stock price charts. These chart patterns play a very important role [38] in technical analysis to reveal the market trends.

Different chart patterns are said to reveal different market trends. For example, a Head-and-Shoulders Tops chart pattern is said to reveal that the market will most likely to have a 20% to 30% rise in the coming future [38]. Identifying this chart pattern is crucial in positioning the investment move. Figure 5.1 shows 16 samples of typical chart patterns.

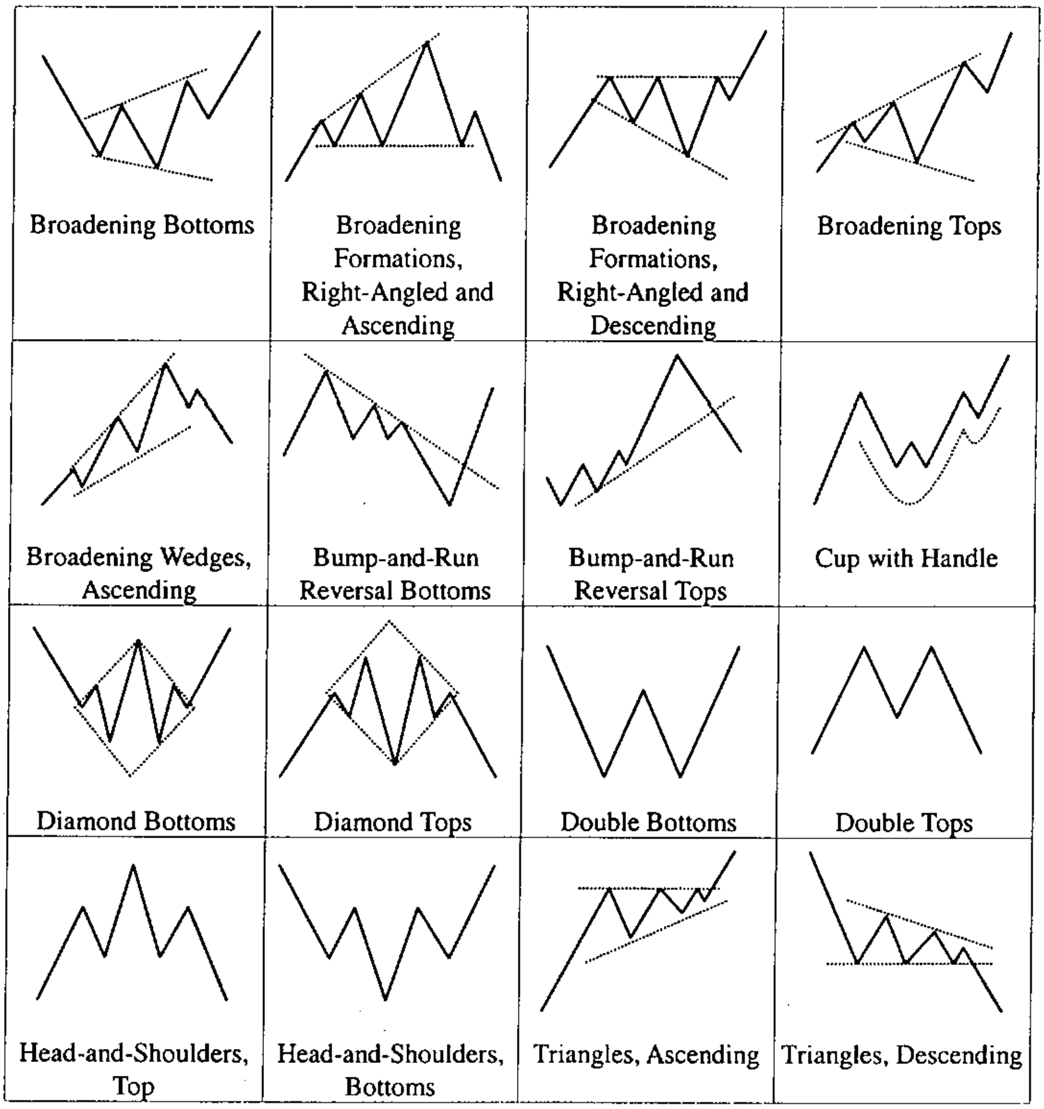


Figure 5.1 – Sixteen Samples of Typical Chart Patterns

5.2 CBR Methodology

The proposed agent incorporates both technical and fundamental analysis to predict the stock market movement, CBR system is studied to address the problem. The advantage of using CBR system but not other intelligent systems such as neural network, expert system and fuzzy logic is that it does not require the presence of strong domain theory. Also, users can have a better understanding of the system behaviour.

5.2.1 Case Representation and Adaptation

In the proposed CBR system, seven features have been selected to represent a case. These features are time range *t*, P/E ratio *p*, average turnover *o*, RSI *r*, OBV *v*, prices moving average *m* and the chart pattern *c*. A feature vector F_i defined as $F_i = \{t, p, o, r, v, m, c\}$.

When an attribute in feature F_i taking value v_1 in case 1 and v_2 in case 2, $v_1, v_2 \in [a,b]$, the similarity of v_1 and v_2 is computed by the following expression:

$$sim(v1, v2) = 1 - \frac{|v1 - v2|}{b - a} \quad \text{for } b \neq a$$

For the attribute 'chart pattern' in feature F, the similarity measure of the attribute between the two cases is measured by the following expression. The extraction of the chart pattern *c* in the time series data is via the use of algorithm

PXtract, which will be discussed in section 5.3.

$$sim(v1, v2) = \begin{cases} 1 & if\ v1 = v2 \\ 0 & otherwise \end{cases}$$

The overall similarity between two cases c_1 and c_2 is measured by the weighted-sum metric shown below:

$$sim(c1, c2) = \frac{\sum_{i=1..n} w_i \cdot sim(v_{1i}, v_{2i})}{\sum_{i=1..n} w_i}$$

The parameter w_i should be arbitrary defined by users. It mainly defends on how the users judged and weighted different technical analysis techniques. For example, if a user have much more trust on 'Chart Patterns Analysis' than others techniques, the weight of 'Chart Patterns' should be much larger than the others.

5.2.2 Case Retrieval

The system retrieves the updated stock data and converts them into case knowledge, then studies the new current status of the cases and appends the new result set into the result database that for users' direct query. The system is set to refer to 3 Successive Cases within a series of stock cases as one complete CASE for prediction purpose.

5.3 Wave Patterns Identification

Analysts use chart patterns in attempt to put all buying and selling into perspective in the hope of concisely illustrating the forces of supply and demand. However, figuring out which underlying pattern in daily price movement for prediction purpose is a non-trivial task.

The analysis and identification of wave pattern is difficult for two reasons. First, no single time scale works for all analytical purposes. Second, for any stock chart, there are countless different pattern combinations, some containing sub-patterns. Choosing the most representative presents quite a dilemma. We propose to address the problem using the PXtract algorithm.

5.3.1 Algorithm PXtract

The PXtract algorithm extracts wave patterns from stock price charts based on the following phases:

5.3.1.1 Window Size Phase.

As there is hardly a single time scale that will work for all analytical purpose in wave identification process [2], a set of time window sizes $W = \{w_1, w_2, \ldots, w_n\} \mid w_1 > w_2 > \ldots > w_n$ is defined (w_n is the windows size). Different window sizes are used to discover whether wave pattern occurs in a specific time range or not. For example, in short-term investment strategy, a possible window size can be defined as $W = \{40, 39, 38, \ldots, 12, 11, 10\}$.

5.3.1.2 Time Subset Generation Phase.

Stock price trading data contains a set of time data $T = \{t_1, t_2, \ldots, t_n\} \mid t_1 > t_2 > \ldots > t_n$. For a given time window size w_i, set T contains a number of subset T, $T' \subset T$.

A Set P is also defined, where $P \subset T$. It contains the time ranges which previously identified wave pattern have occurred, set P is ϕ in the beginning.

It is said that any large change in the trend plays a more important role in the prediction process. The range which has been previously discovered to contain a wave pattern will not be tested again (i.e. $T' \not\subset P$). Details of the time subset T' generation process are shown in Figure 5.2.

current window size = w
$T = \{ t_1, t_2, \ldots, t_n \}$
for (k=1; $k+w<n$; k++) **do**
  begin
  $S = (t_k, t_{k+1}, t_{k+2}, \ldots, t_{k+w})$
  if ($S \not\subset P$)
    $T' = T' \cup S$
  end
return T'

Figure 5.2 – Time Subset Generation

For example, T = {10 Jan, 9 Jan, 8 Jan, 7 Jan, 6 Jan, 5 Jan, 4 Jan, 3 Jan, 2 Jan, 1 Jan}, current testing window size is 3 (w = 3), P = {9 Jan, 8 Jan, 7 Jan, 6 Jan}. After the time subset generation process, T' = {(5 Jan, 4 Jan, 3 Jan), (4 Jan, 3 Jan, 2 Jan), (3 Jan, 2 Jan, 1 Jan)}

5.3.1.3 Pattern Recognition.

For a given set of time $T' \mid T' \subset T$, apply the wavelet theory to identify the desired sequences. If a predefined wave pattern is discovered, add T' to P. Details are described in section 5.2.2.

foreach w_i in W ***do***

$T' = genSet(w_i)$ // $T' \not\subset P$

foreach T'' in T' **do**

if (T'' contains Chart Pattern)

$P = \cup T''$

end

End

Figure 5.3 – Algorithm PXtract

The proposed algorithm PXtract is given in Figure 5.3. The function *genSet*(w_i) is the subset generation process discussed in section 5.2.1.2. At the end of the algorithm, all the time information of the identified wave pattern will be stored in set *P*.

5.3.2 Wavelet Recognition

Wavelet analysis has been applied widely and with encouraging results in signal processing, image processing and pattern recognition [44]. Stock charts waves are 1-D patterns. Obviously, then, it is not necessary to transform them from a higher dimension to 1-D.

Wavelet analysis is a recent development of applied mathematics in 1980s by J.Morlet. It denotes a univariate function ψ, which is defined on $\Re$ when subjected to fundament operations of shifts and dyadic dilation, yields an orthogonal basis of $L^2(\Re)$.

Multi-resolution analysis (MRA) was first published in 1989 by Mallat and Meyer [32]. It was formulated based on the study of orthonormal, compactly supported wavelet bases.

The wavelet basis induces a MRA on $L^2(R)$, the decomposition of the Hilbert space $L^2(R)$, into a chain of closed subspaces.

$$\cdots \subset V_4 \subset V_3 \subset V_2 \subset V_1 \subset V_0 \subset \cdots$$

MRA can be applied to the function $f(x)$ and decompose it to $L^2(R)$ space. In MRA, closed sub-space V_{j-1} can be decomposed orthogonally as:

$$V_j = V_{j+1} \oplus W_{j+1} \qquad (5.1)$$

V_j contains the low-frequency signal component of V_{j-1} and W_j contains the high frequency signal component of V_{j-1}. According to the wavelet orthonormal decomposition as shown in equation 5.1, first V_j is decomposed orthogonally into a high-frequency sub-space V_{j+1} and W_{j+1}. The low-frequency sub-space V_{j+1} is further decomposed into V_{j+2} and W_{j+2} and the processes can be continued. The above wavelet orthonormal decomposition can be represented by

$$
\begin{aligned}
V_j &= W_{j+1} \oplus V_{j+1} \\
&= W_{j+1} \oplus W_{j+2} \oplus V_{j+2} \\
&= W_{j+1} \oplus W_{j+2} \oplus W_{j+3} \oplus V_{j+3} \\
&= \ldots..
\end{aligned}
$$

According to the wavelet orthonormal decomposition as shown in equation 5.1, first V_0 is decomposed orthogonal into a high-frequency sub-space W_0 and a low-frequency one V_0 by using the wavelet transform equations.

In the Chart Pattern Recognition process, V_0 should be the original wave pattern, while V_1 and W_1 should be the wavelet transformed sub-pattern.

If we want to analyze the current data to determine whether it is a predefined chart pattern, a template of the chart pattern is needed. According to the noisy input data, direct comparing the data with the template will lead to an incorrect result. Therefore, wavelet decomposition should be applied to both the input data and the template. Example of matching the input data to a 'Head and Shoulder, Top' pattern is illustrated in Figure 5.4.

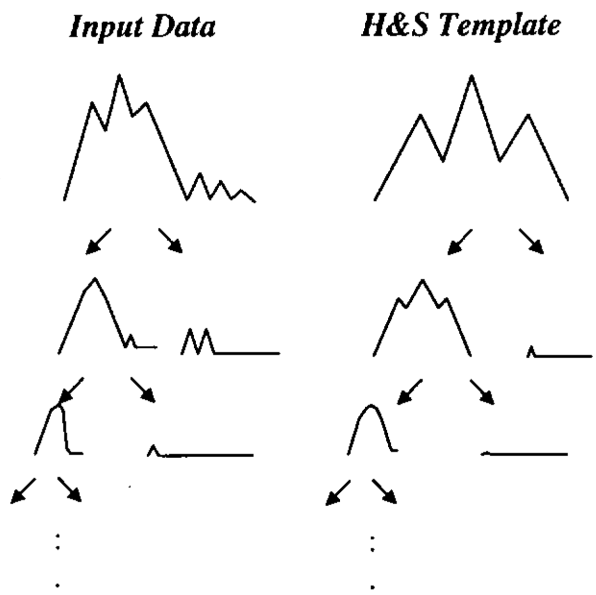


Figure 5.4 – Wavelet Decomposition in Both Input Data and Chart Pattern Template

5.3.3 Matching

We can match input data and the template using either of two methods: Simple Multi-resolution (MR) Matching and Radial Basis Function Neural Network (RBFNN) matching.

5.3.3.1 Simple Multi-resolution (MR) Matching

We can match sub-patterns using either a range of coarse to fine scales, matching the input data with features in the pattern template. The matching process will only be terminated if the target is accepted or rejected. If the result is undetermined, it continues to at the next, finer scale. The coarse scale coefficients obtained from the low pass filter represent the global features of the signal.

For a high resolution scale, the intraclass variance will be larger than the low one. A threshold scale should be defined for the determination of the acceptance. For example, scale n is defined as the lowest resolution. The *resolution threshold* is t and t > n. At each resolution t, its root-means-square should be greater than another threshold value l, which named as *level threshold*.It is difficult to derive optimal thresholds, therefore we need to determine this through empirical testing. Figure 5.5 illustrates the details of the process.

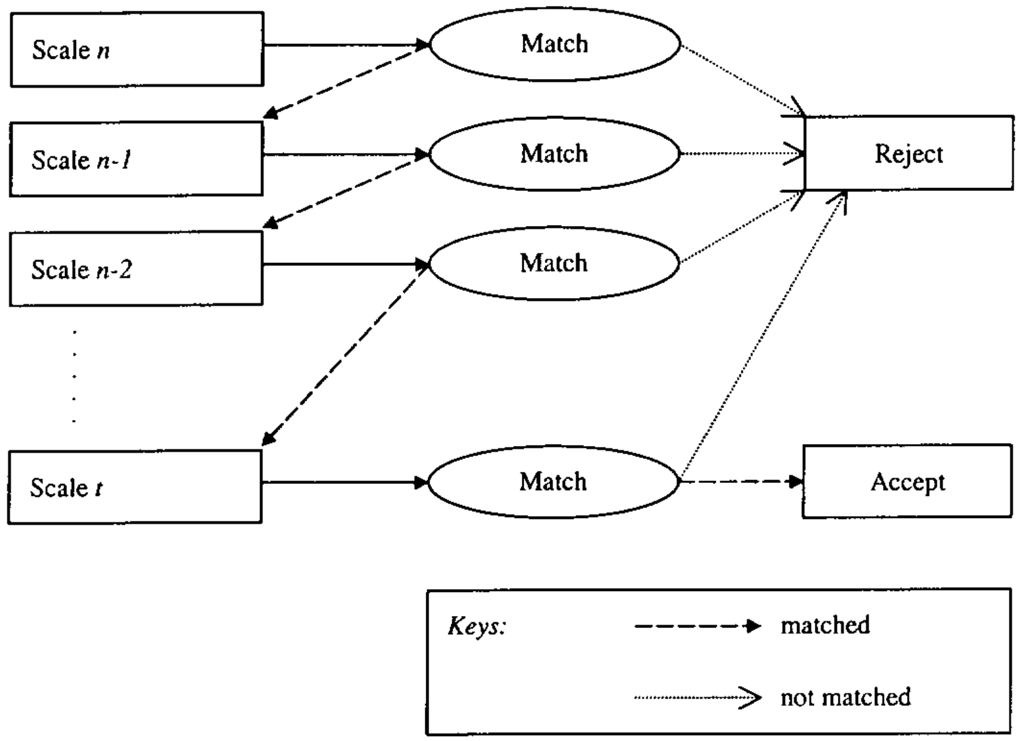


Figure 5.5 – The Multi-resolution Matching

5.3.3.2 Radial Basis Function Neural Network (RBFNN) Matching

Neural networks are widely used in pattern recognition and classification problems. Radial basis function neural network (RBFNN), its universal approximation capabilities have been proved by Park and Sandberg [34, 33] as quite appropriate for solving our pattern/signal matching problem [4]. We have created different RBFNNs for recognizing different patterns at different resolution levels. The input of the network is the wavelet transformed values in a particular resolution. Figure 5.6 illustrates the architecture of the RBFNN.

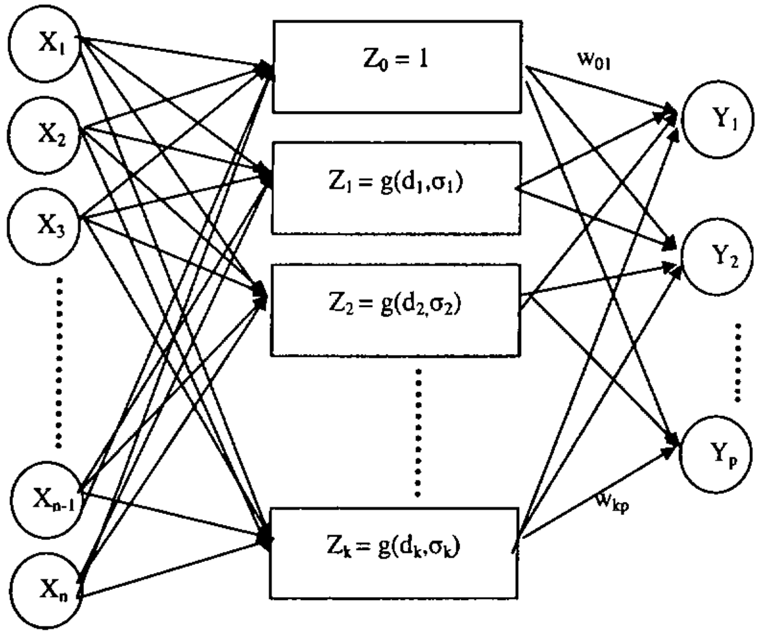


Figure 5.6 – Structure of RBFNN for Chart Pattern Recognition

The network contains three layers and each layer is made up of several nodes. The hidden layer, which contains K nodes, processes the input information with a nonlinear transformation. The nonlinear transformation function is implemented by using the Gaussian function:

$$Z_j(x) = \exp\left(\frac{\| x - d_j \|^2}{2\sigma_j^2}\right) \tag{5.2}$$

where $Z_j(x)$ is the output of the j^{th} hidden node; $Z_0(x)$ is the bias node, fixed at value of 1; x is the input with n-dimensions; d_j and σ_j are the center and variance of the j^{th} hidden node respectively. The distance between the center of the hidden node and the input is measured by Euclidean distance. The total number of output nodes is P and the output value is a linear combination of the nonlinear RBF $Z_j(x)$

$$Y_p(x) = \sum_{j=0}^{J} w_{jp} Z_j(x) \tag{5.3}$$

where $Y_p(x)$ is the p^{th} output node and w_{jp} is the weight of the connection between the j^{th} hidden node and the p^{th} output node

To simplify the network, the choice of the centers of the Gaussian functions in equation 5.2 is determined by the K-means algorithm [39]. The variances of the Gaussians σ_j are chosen to be equal to the mean distance of every Gaussians center from its few neighboring Gaussian centers. A constructive learning approach is used to select the number of hidden units in the RBF network. The hidden nodes are created one at a time. At each iteration, it adds one hidden node and checked for the error of the new network. This procedure is repeated until the error goal is met, or the preset maximum number of hidden nodes is reached.

Stock chart pattern identification is highly subjective and humans are far superior to machines in recognizing which stock patterns are meaningful to investors. Over the period of 1 January 1995 to 31 December 2001, we studied five different stocks (Table 5.1). Following the rules suggested in [38] whereby the judgment of a human critic, we extracted the 14 different chart patterns as shown in Figure 5.1. In the training set, there have totally 231 training inputs (a quarter of 308 pattern templates used as validation set) for the network. We initially set the wavelet resolution equal to 8. We found that the signal/pattern for the resolution 1 to 3, was too smooth and patterns were similar to each others at those levels. The network could not recognize different patterns well. Therefore, only 4 RBFNNs were created for training different chart patterns at the resolution levels 4 – 7. The performance of the networks at different resolution levels and the classification results are discussed in the next section.

5.4. Training Set Collections

Extracting chart patterns in the stock time series data is a time consuming and expensive operation. In our training set, only 64 real data for 14 different chart patterns had been collected from 5 different stocks. But it is insufficient for training the system well. If we atempted to extract over 200 chart patterns in the time series data, it would be infeasible, time consuming and expensive. In order to expand the training set, we adopted a simple but powerful mechanism to generate more training data based on the real data.

To generate more training samples, The *Radial Deformation* method is used.

Here is the major step of the radial deformation process:

a) $P = \{p_1, p_2, p_3, \ldots, p_n\}$is a set of data points containing a chart pattern.

b) Randomly picking i points ($i <= n$) in set P for deformation.

c) Randomly generated a set of the *radial deformation distance* $D = \{d_1, d_2, \ldots, d_i\}$.

d) For each point in P, a random step d_r is taken along a random direction. The deformed pattern is constructed by joining consecutive points with straight lines. Details are depicted in Figure 5.7.

e) Justify the deformed pattern by human critics.

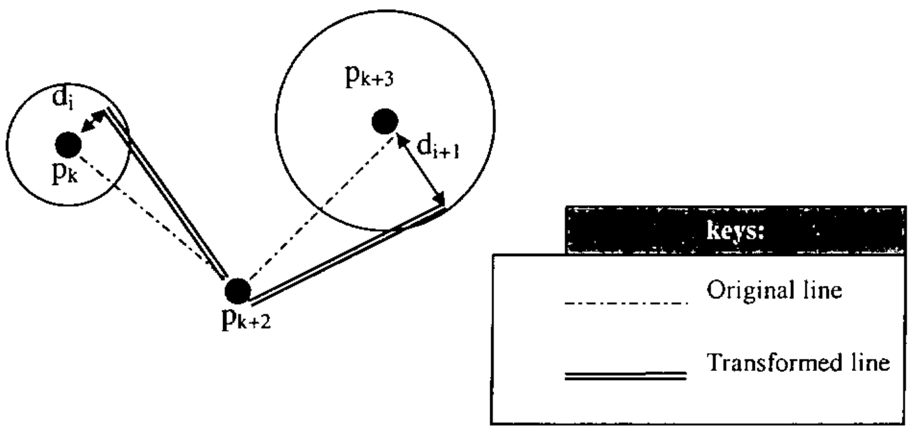


Figure 5.7 Radial Deformation

Psychophysical studies [43] tell us that humans are better than machines at recognizing object which are meaningful to humans. For the assessment of the generated training data, is whereby the judgment of human critics, the whole training set (including real and generated data) is accepted and selected from the human critic.

In the training set, 64 chart patterns had been extracted from 5 different stocks in the Hong Kong stock market. Table 5.1 lists the five different stocks. The total numbers of real training data for 14 different chart patterns are shown in Table 5.2 respectively. Details are attached in the Appendix A.

Stock ID	Stock Name
00341	CAFÉ DE CORAL HOLDINGS LTD.
00293	CATHAY PACIFIC AIRWAYS LTD.
00011	HANG SENG BANK LTD.
00005	HSBC HOLDINGS PLC.
00016	SUN HUNG KAI PROPERTIES LTD.

Table 5.1 – The Five Different Stocks and Their Stock IDs

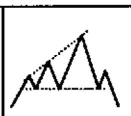
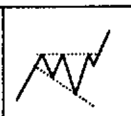
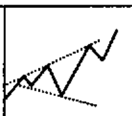
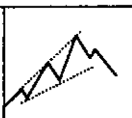
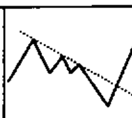
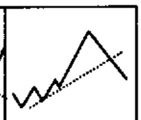
Chart Patterns							
	Broadening Bottoms	Broadening Formations, Right-Angled and Ascending	Broadening Formations, Right-Angled and Descending	Broadening Tops	Broadening Wedges, Ascending	Bump-and-Run Reversal Bottoms	Bump-and-Run Reversal Tops
Total no. of real training data	4	6	3	2	5	8	3

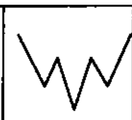
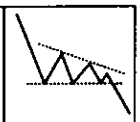
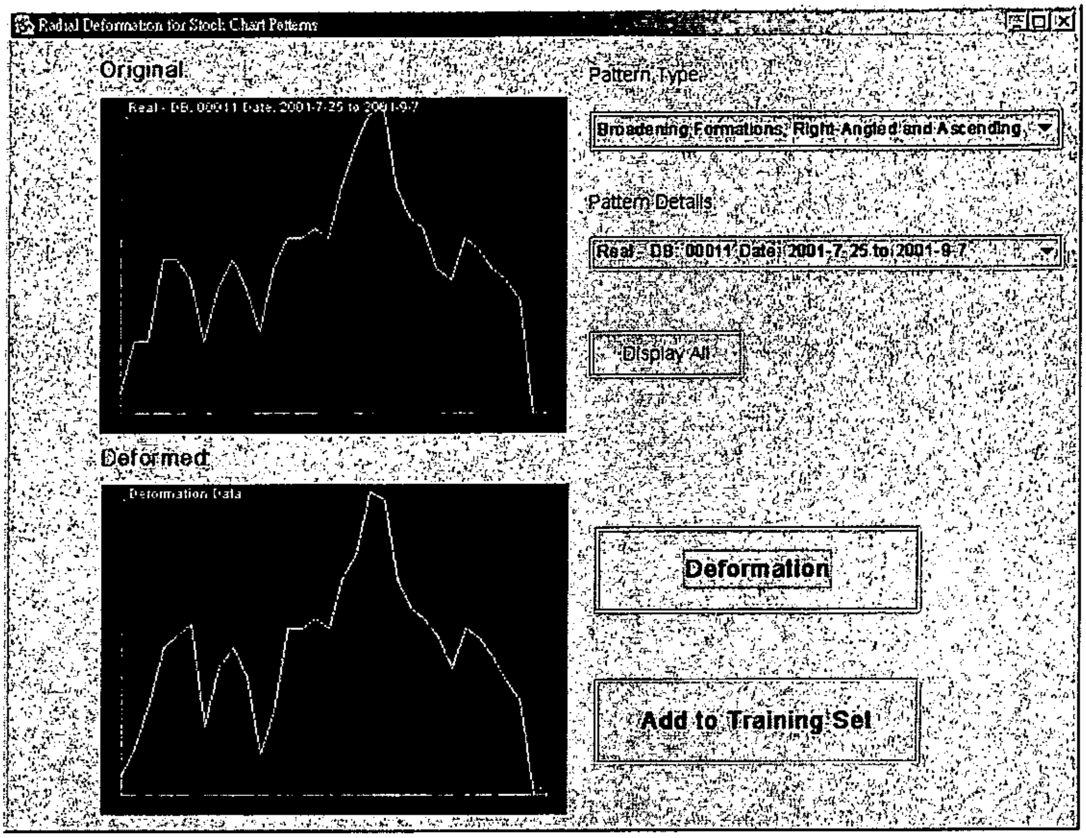
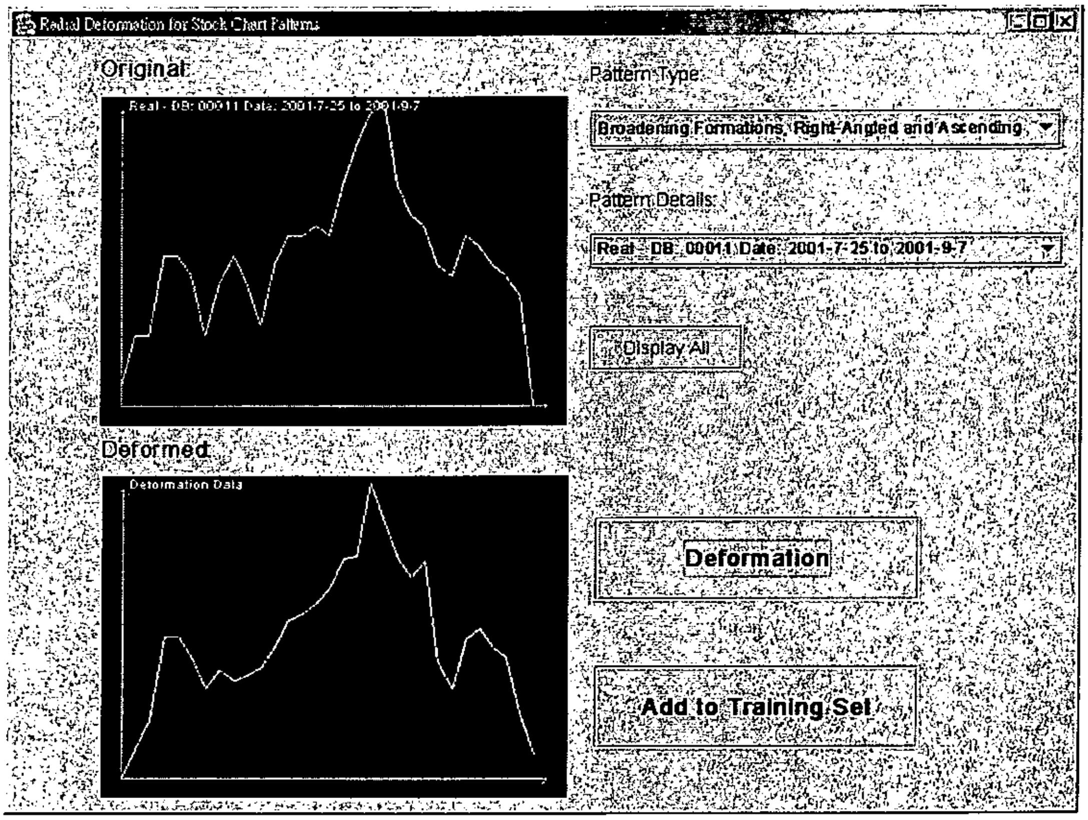
Chart Patterns							
	Cup with Handle	Double Bottoms	Double Tops	Head-and-Shoulders, Top	Head-and-Shoulders, Bottoms	Triangles, Ascending	Triangles, Descending
Total no. of real training data	4	10	5	5	4	2	3

Table 5.2 – Total Numbers of Training Patterns in Fourteen Typical Chart Patterns of Five Different Stocks.

By applying the Radial Deformation technique, the 64 real training patterns had been extended to totally 308 patterns. All the patterns generated by Radial Deformation must be judged by human, to determine whether the deformed pattern is accepted, meaningful to human or not. Figure 5.8 illustrates examples of (a) accepted and (b) NOT accepted deformed patterns. Details of all the deformed patterns can be found in Appendix B.



a) Example of Accepted Deformed Pattern



b) Example of NOT Accepted Deformed Pattern

Figure 5.8 – Accepted and NOT Accepted Deformed Pattern by Radial Transformation.

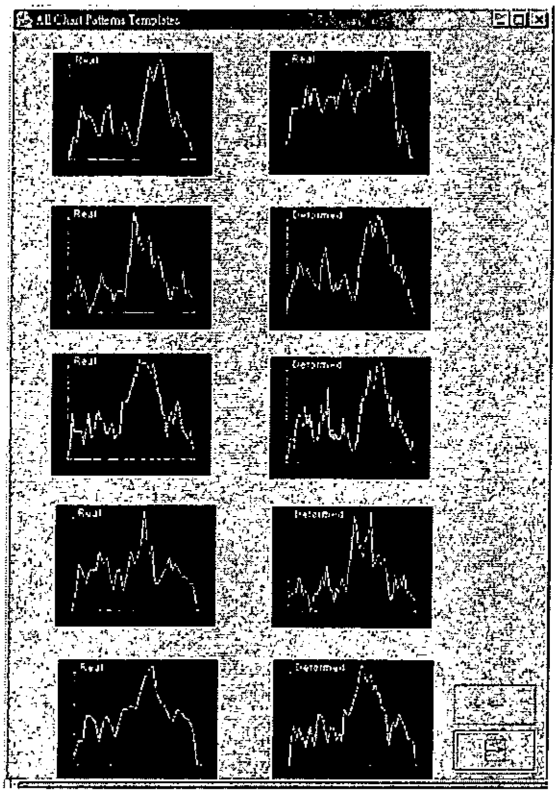


Figure 5.9 – Chart Pattern Training Set from Both Real and Deformed Chart Patterns.

5.5. Experiments

Two set of experiments have been taken to evaluate the accuracy of the proposed system. The first set evaluates whether the algorithm PXtract is scaleable, and the second test evaluates the performance comparison between using simple multi-resolution matching and RBFNN matching

Algorithm PXtract uses different time window sizes to locate any occurrence of a specific chart pattern. The major concern of the algorithm is the performance. To assess the relative performance of the algorithms and investigate their scale-up properties, we performed experiments on an IBM PC workstation with a Pentium 3 500MHz CPU, 128MB memory

To evaluate the performance of the algorithm using RBFNN Matching over a large range of widow's size, typical stock prices of SUN HUNG KAI & CO. LTD. (0086) for the period from 2 Jan 1992 to 31 Dec 2001 were used

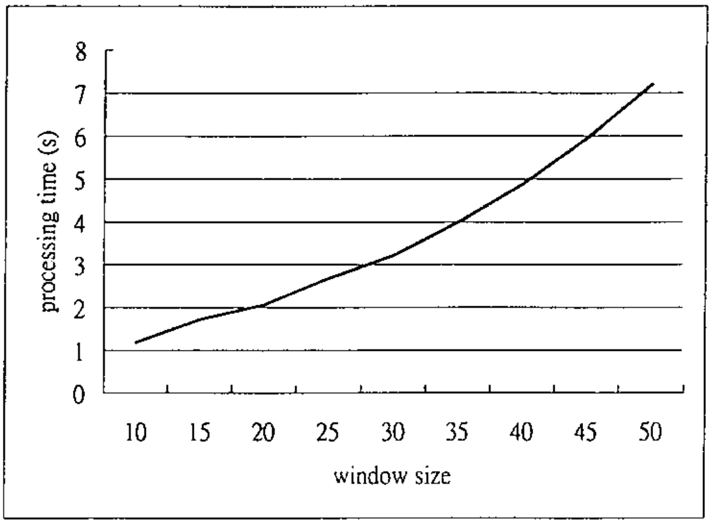


Figure 5.10 – Execution Time of Algorithm PXtract Using RBFNN Matching in Different Time Window Sizes

It is found that the algorithm performance scale linearly with increasing the time window sizes.

In the experiments of wavelet chart patterns recognition, different wavelet families are selected as the filter. The maximum resolution level was set to be 7. The highest resolution level 8 is taken as the raw input. Fig. 5.11 illustrates the multi-resolution recognition process.

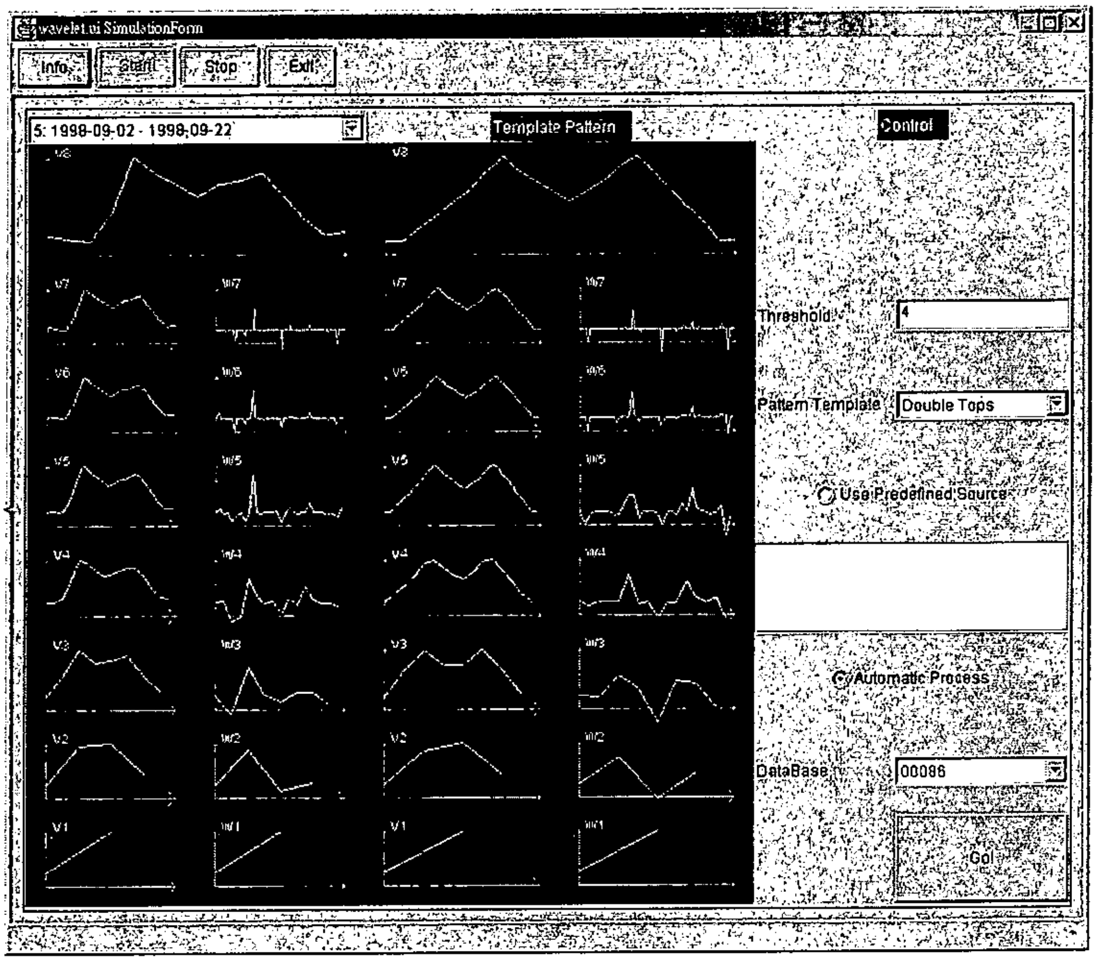


Figure 5.11 Algorithm PXtract Using Wavelet Multi-resolution Analysis on the Pattern 'Double Tops' Template

On the left hand side of Fig. 5.11, it is the stock price of the stock SUN HUNG KAI & CO. LTD. (0086) for the period from 2 September 1998 to 22 September 1998. This period contains the 'Double Tops' pattern. For the identification of the chart patterns, two matching methods were studied and they are simple multi-resolution (MR) Matching and RBFNN Matching. For the simple MR matching, similarity between the input and the template is measured by mean absolute percentage error (MAPE). A low MAPE means that they are similar. In order to examine the performance of the Simple MR Matching, experiments using different resolution threshold *t* and different level threshold *l* had been carried out. Experiments had been carried out which using different wavelet families, resolution threshold *t*, level threshold *l*. Table 5.3 shows the most accurate combinations. Detail results of different setting are attached in Appendix C.

In this experiment, we found that the accuracy using Simple MR Matching is not accurate with an average recognition rate of 30%. Furthermore, the calculation of MAPE between the input data and the pattern templates is a heavy workload. Although the recognition rate of more than 50% can be reached, if we set the level threshold at a low value (about 0.1) and high resolution threshold (above 6), the processing time is unacceptable long (about 3143s). This illustrates that Simple MR Matching is not a good choice in the matching process.

Wavelet Family	Resolution threshold	Threshold value	Accuracy	Total number of patterns discovered	Processing Time
Daubechies (DB2)	4	0.3	6.2%	8932	312s
		0.2	7.1%	7419	
		0.15	14.2%	3936	
		0.1	43.1%	543	
	5	0.3	7.1%	7734	931s
		0.2	9.4%	6498	
		0.15	17.4%	2096	
		0.1	53%	420	
	6	0.3	8.9%	7146	3143s
		0.2	13.5%	5942	
		0.15	19.9%	1873	
		0.1	56.9%	231	
	7	0.3	10.5%	6023	8328s
		0.2	14.5%	5129	
		0.15	18.5%	1543	
		0.1	48.3%	194	

Table 5.3 – Optimal Wavelet and Thresholds Setting Found by Empirical Testing

In the experiment of RBFNN matching, 4 RBFNNs have been created for the resolution levels of 4 – 7. The training set of the chart pattern is collected by the judgment of a human critic following the rules suggested in [38] from the real and deformed data mentioned in Section 5.4.

The training set contains totally 308 records. A quarter of the training set is extracted as the validation set. Table 5.4 illustrates the overall classification results. It shows that the classification rate is over 90% and the optimal recognition resolution level is 6. Four wavelet families had been tested, and their performance were more or less the same except that the Haar wavelet was not suitable for use.

Wavelet Families	Resolution level	Training set	Validation set
Haar DB1	4	66%	64%
	5	75%	72%
	6	81%	78%
	7	87%	74%
Daubechies DB2	4	73%	64%
	5	85%	78%
	6	95%	91%
	7	97%	85%
Coiflet C1	4	77%	72%
	5	86%	81%
	6	95%	90%
	7	98%	84%
Symmlet S8	4	75%	68%
	5	84%	78%
	6	93%	89%
	7	96%	82%

Table 5.4 – Accuracy of Different RBFNNs in Different Wavelet Families and Resolution Levels

After we found the appropriate setting of the RBFNN, we applied it to extract all the chart patterns from data of 10 different stocks over a 10-year period. Table 5.5 shows the accuracy of the 14 different chart patterns. The RBFFNN has an average accuracy of 81%.

Chart Pattern	Accuracy
Broadening Bottoms	73%
Broadening Formations, Right-Angled and Ascending	84%
Broadening Formations, Right-Angled and Descending	81%
Broadening Tops	79%
Broadening Wedges, Ascending	86%
Bump-and-Run Reversal Bottoms	83%
Bump-and-Run Reversal Tops	82%
Cup with Handle	63%
Double Bottoms	92%
Double Tops	89%
Head-and-Shoulders, Top	86%
Head-and-Shoulders, Bottoms	87%
Triangles, Ascending	73%
Triangles, Descending	76%

Table 5.5 – Accuracy of Identifying the Fourteen Different Chart Patterns using RBFNN Extraction Methods

Multi-resoultion RBFNN Matching has a high accuracy in recognizing different chart patterns. However, the accuracy of the recognition process is heavily dependent on the resolution level. Once the pattern has been identified, based on empirical testing, the proposed method is highly accurate.

Chapter 6
Adaptive Agent

Artificial neural network is widely used in solving classification, pattern identification and prediction problems. Stock market forecasting is an example of a particularly complex forecasting problem in which non-linear forecasting methods such as artificial neural networks have been successfully employed [16, 21]. But the prediction and classification performance of neural networks critically depend on the optimal choice of parameters such as transformations and selection of raw data, network structure and the output format. Weinstein [41] stated that every stock has its own characteristic; different model should be built in order to adapt different characteristics. For example, to forecast the stock prices of the stock "CHINA PETROLEUM & CHEMICAL CORPORATION HK" which categorized as one of the 'Oil and Resources' stocks, information such as the oil prices may need to be considered. In contrast, for the stock "NEW WORLD CYBERBASE LTD HK" which categorized as one of the "Software and Service" stocks, information such as oil prices may need not be considered. Since the topology of nodes and connections in the hidden layer depends on the number of input data and it is stock dependent, different neural networks with different structures should be built for different stocks.

Since there is no known analytic solution, genetic searching approach is proposed to solve these optimization problems. Genetic algorithm is one of the evolution algorithms that model biological processes, constitute a suitable optimization method and it performs well on large, non-linear searching spaces.

6.1 Input Value Selection

Weinstein [41] found that every stock has its own characteristics. It mainly falls into five categories, they are: Finance, Utilities, Property, and Commercial/Industrial and Technology. Stocks' price movements in different categories are depending on different factors. It is difficult to identity which set of factors will affect a particular stock's price movement. Genetic algorithm provides a dynamic mechanism in addressing the input selection problem.

Input for time-series forecasting can be divided into two groups. They are fundamental data and technical Indicators.

6.1.1 Fundamental Data

Fundamental data is the raw data that is public known; everyone can obtain through the mass media. Many of financial researchers believe that there have some hidden indicators and patterns underlying it [37]. This study, focuses on the HK stock market in which, 13 potential input parameters have been investigated, these include: daily high, daily low, daily opening, daily closing, daily turnover, gold price, oil price, exchange rate between HK to US, HK deposit call, HK interbank call, HK prime rate, silver price and Hang Seng Index.

Hang Seng Index is a barometer of the Hong Kong stock market. The constituent stocks are grouped under Commerce and Industry, Finance, Properties and Utilities sub-indexes. The constituent of the stocks comprises 33 stocks, which are representative of the market. The aggregate market capitalization of these stocks accounts for about 79% of the total market capitalization on The Stock Exchange of Hong Kong Limited (SEHK). So, it is a market-value weighted index of the stock prices of the 33 largest companies in the Hong Kong stock market.

6.1.2 Technical Indicators

There have a lot of popular technical indicators, the most popular indicators are Relative Strength Index (RSI), Moving Average (MA), Stochastic and Ballinger Bands [9]. Each of them provides guidance for investors to analyze the trend of the stocks' prices movements. In this study, two of the popular indictors, RSI and MA are chosen as one of the potential inputs for training the networks.

6.2 Data Selection

Degree of prices/indexes movements is one of the important signals for forecasting the market movements [35]. In order to identify the effect, two transformations of the fundamental data which discussed in section 6.1.1 have also been studied, they are simple difference and percentage difference.

Equations 6.1 and 6.2 show the calculation of simple difference and percentage difference respectively.

$$\Delta(t,t-1) = price(t) - price(t-1) \qquad (6.1)$$

$$percentage\ (t,t-1) = \frac{\Delta(t,t-1)}{price(t-1)} \qquad (6.2)$$

where *price(t)* and *price(t-*1) are the stock price at time *t* and time *t*-1 respectively.

Time lags [3] is also an important issue in financial forecasting. It is better to consider several trading days' prices movements instead one. In this study, time window size is set to be three, thus stock price in trading day *t price(t)*, *t* –1 *price(t-*1*)* and *t*-2 *price(t-*2*)* are also taken as potential inputs in the network.

The total number of potential inputs to be tested is fifty-seven and the testing result is shown in Table 6.1. Details of the tests will be discussed in section 6.4.1.

6.3 Network Architecture

Determining what to be forecasted is a critical component of success. People are more concern on how much profit they can earn rather than to know the forecasted prices value in the following trading days. Thus, the system should be able to provide suggestions on one of the three investment strategies: buy, hold or sell to the investors.

'Buy Low, Sell High' is the golden rule for all investors. For most investors, if one foresees that the stock prices will have a certain degree of upward movement, he will buy the stocks. In contrast, if one foresees that a certain degree of drop will happen, he will sell his stocks.

The system models this human behavior, if it forecasts an upward or downward prices movement which will reach to a threshold T, it will give advise to buy or sell respectively, and hold otherwise. The threshold T is defined as *decision threshold*, its effects will be discussed in section 6.3.3.

6.3.1 Network Inputs

Multi-layers backward propagation neural network is used as the network architecture. A good input to the networks is one of the most important factors to get a right prediction/classification. Too many inputs will cause the network over-fitting, filtering the input variables is necessary.

The selection of inputs is based on genetic algorithm, each potential input as a gene in a chromosome, encoded as a binary string in which '1' represents the adoption of the variable and '0' represents no adoption. The chromosome is shown in Fig. 6.1.

Var1	Var2	Var3	Var4	...
1	0	0	1	...
Gene1	Gene2	Gene3	Gene4	...

Figure 6.1 – Variables and Values Representation as a Chromosome

Each chromosome associated with a neural network. Its cost/fitness function is based on the network's accuracy. Higher accuracy will have greater fitness value. Since the network architecture (number of layers and hidden nodes) has not been determined at this moment, we assume that the initial network has only one hidden layer and the number of hidden nodes can be derived by equation 6.3 as suggested in [42]. Genetic algorithm will then be used to optimize the network architecture.

$$\text{number of hiden nodes} = \frac{P}{(10 + (m+n)} \qquad (6.3)$$

where P is the number of training pattern, n and m are the number of inputs and outputs respectively.

6.3.2 Network Topology

The number of hidden nodes in the network affects its prediction and classification ability. Generally, too simple network architecture is unable to extract high order, non-linear features. The more number of hidden nodes and layers, the more complicated pattern it can recognize. However, if the network structure is too complex, it is said to just memorize the specific features but not learn its underlying features.

Fahlman [10] proposed a Cascade Correlation Algorithm which addresses the problem by increasing the number of neurons and layers until over-learning occurs. However, it requires a good starting solution. If the starting network is too simple, it will be unable to recognize higher order features, and optimization may fail. As finding a good starting solution is non-trivial, cascade correlation algorithm may not be a good solution for finding an optimal topology.

Since genetic algorithm starts from different, random topologies, it has the capability to find the global optimal network topology for system. The cost/fitness value is calculated by accuracy of the trained network using the input selected in section 6.3.1. The chromosome contains 14 bits that represent some typical network architecture as shown in Fig. 6.2. The first two bits represent how many hidden layers it has. Up to three hidden layers can be represented by using two bits string. The rest three groups of four bits string represent how many hidden nodes in the first, second and third hidden layers respectively.

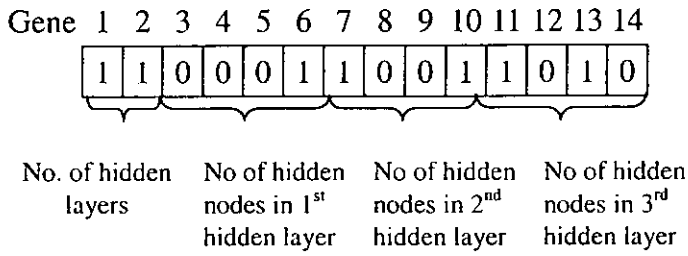


Figure 6.2 – Chromosome Representing a Typical Network Architecture

Since the first 2 genes give the number of hidden layers the network has, if we apply no constraints in the genetic operations, useless operations may take place. E.g. for a chromosome which just has one hidden layer and crossover had been taking in gene 8, no new information can be generated since genes 1-7 have no changes. Constraints should be applied in this scenario to avoid such useless operation. Several constraints have been applied in this study, they are:

1. Genes 3-6, 7-10 and 11-14 cannot all be equal to zero.
2. Crossover and mutation cannot occur in genes 1 and 2.
3. If no. of hidden layers in the chromosome is 1, crossover and mutation should occur between genes 3-6.

If no. of hidden layers in the chromosome is 2, crossover and mutation should occur between genes 3-10.

6.3.3 Network Outputs

Determining what is to be forecasted is very important. Suppose we want to forecast tomorrow's closing price of Dow Jones Industrial Average (DJIA). If the DJIA is approximately 3800 and it changes each day by about 10 points, then this change corresponds to about 0.263 percents of the given value. If we intend to forecast the DJIA to an accuracy of plus or minus 2 DJIA points, then we require an accuracy with 0.05 percent error (2 points of 3800). It is nearly infeasible.

The only three investment strategies are: buy, hold and sell. It is well known that the golden rule in stock market is 'Buy low, Sell high'. In this study, we defined the best strategy in trading time t as:

$$\text{Best Strategy} = \begin{cases} \text{Buy} \quad \text{if } \dfrac{price(t+1) - price(t)}{price(t)} > z\% \\ \text{Sell} \quad \text{if } \dfrac{price(t+1) - price(t)}{price(t)} < z\% \\ \text{Hold} \quad \text{otherwise} \end{cases} \tag{6.4}$$

where z is the *decision threshold*. *Decision threshold* is defined in order to help the system transforms the forecasting problem into a classification problem. Section 6.4.3 will discuss how the *decision threshold* represents the risk. *price(t)* and *price(t+1)* are the stock prices at present and next trading day respectively.

The output of the network is encoded as 1, 0 and –1, which represent the suggested investment strategies 'buy', 'hold' and 'sell' respectively.

6.4 Experiments and Evaluations

Three experiments have been carried out. The first is to find out which input parameters are optimal for different stocks' model in Hong Kong. The second test is to find out the best network architecture for different models. The last experiment is a simulation test; simulating the investment processes in a particular period. Total gain is given out as a reference of how much advantage that the investors can take if one just follows the system suggestion.

6.4.1 Experiment A

In this experiment, genetic algorithm is applied to optimize the network input. This study has used 15 different input parameters, they are: daily high, daily low, daily opening, daily closing, daily turnover, gold price, oil price, exchange rate between HK to US, HK deposit call, HK interbank call, HK prime rate, silver price, Hang Seng Index, Relative Strength Index and Moving Average.

Two transformations of the 15 parameters have also been taken into consideration. They are simple difference (Δ) and percentage difference (%) which have been discussed in Section 6.2.3.

The fitness value of the chromosome in the genetic algorithm is the network's classification rate. Different values of decision threshold z that discussed in section 6.3.3 will affect the result. Two decision thresholds are tested in this study, the first one is equal to the percentage change in average price (abbreviated as *avgMov*) and the other is the *avgMov* / 2. The test results are shown in Table 6.1.

In Table 6.1, input with 'X' means it is selected by setting the decision threshold to *avgMov*, 'O' represents using decision threshold with *avgMov*/2. It is observed that the daily closing price and its transformation is the most frequent used input. In contrast, technical indicators such as RSI and MA are rarely used. It can conclude that technical indictors are not a necessary component in the network input. Also, the transformed data is used more frequently than the raw data. Data preprocessing is essential for a good network.

	Stock ID																				Total
	00341		00293		00579		00010		00011		00044		00050		00005		00013		00016		
Daily opening			X	O		O			X												4
Daily closing			X	O	X	O			X			O					X				7
Daily high		O				O													X		3
Daily low		O	X				X			O							X			O	6
Daily turnover		O	X		X					O			X				X			O	7
Castor oil	X											O		O						O	4
Gold					X				X										X		3
HSI low																			X		1
HSI high					X							O	X		X						4
HSI close					X					O									X		3
HSI open	X										X										2
Exchange rate (HK to US)					X														X		2
HK Deposit Call													X								1
HK Interbank Call						O							X			O			X		4
HK Prime rate				O	X		X		X				X		X	O	X	O		O	10
Silver ounce (Mexico)									X	O											2
ΔDaily opening	X	O					X					O			X		X	O		O	8
ΔDaily closing	X	O	X		X	O			X	O	X	O	X	O				O	X	O	14
ΔDaily high	X		X		X	O	X	O		O		O	X		X	O	X		X	O	14
ΔDaily low	X	O			X			O	X						X	O			X		8
ΔDaily turnover			X										X								2
ΔCastor oil		O																			1
ΔGold	X									O							X				3
ΔHSI low																					0
ΔHSI high		O													X			O	X		4
ΔHSI close						O				O						O					3
ΔHSI open						O															1
ΔExchange rate (HK to US)							X	O													2
ΔHK Deposit Call																O					1
ΔHK Interbank Call													X								1
ΔHK Prime rate	X																				1
ΔSilver ounce (Mexico)																					0
%Daily opening			X		X		X			O				O		O					6
%Daily closing	X	O	X		X		X		X		X	O	X	O	X	O	X	O	X		15
%Daily high	X	O	X						X		X	O			X	O			X	O	10
%Daily low		O				O	X			O		O	X	O			X		X		9
%Daily turnover									X												1
%Castor oil	X									O		O								O	4
%Gold			X				X		X								X				4
%HSI low							X				X								X		3
%HSI high	X	O					X	O		O											5
%HSI close					X		X			O							X				4
%HSI open	X						X	O							X						4
%Exchange rate (HK to US)	X											O									2
%HK Deposit Call		O												O	X						3
%HK Interbank Call						O				O								O			3
%HK Prime rate			X														X				2
% Silver ounce (Mexico)							X					O	X								3
price(t)																					0
price(t-1)								O							X						2
price(t-2)																			X		1
9 days RSI																					0
14 days RSI											X										1
10 days MA													X		X						2
20 days MA									X												1
30 days MA								O										O			2
Total numbers of input: (Threshold $z = avgMov$)	14		12		13		14		12		6		13		12		12		15		123
Total numbers of input: (Threshold $z = avgMov/2$)	13		3		10		7		14		12		6		9		7		9		90

Table 6.1 – Different Inputs Selected for Different Networks by Genetic Algorithm

Stock ID	Stock Name
00341	CAFÉ DE CORAL HOLDINGS LTD.
00293	CATHAY PACIFIC AIRWAYS LTD.
00579	CHEVALIER CONSTRUCTION HOLDINGS LTD
00010	HANG LUNG GROUP LTD.
00011	HANG SENG BANK LTD.
00044	H.K. AIRCRAFT ENGINEERING CO. LTD.
00050	HONG KONG FERRY (HOLDINGS) CO. LTD.
00005	HSBC HOLDINGS PLC.
00013	HUTCHISON WHAMPOA LTD
00016	SUN HUNG KAI PROPERTIES LTD.

Table 6.2 – Stock ID and Its Name

Stock ID	Decision threshold ($z = avgMov$)			Decision threshold ($z = avgMov/2$)		
	Network architecture	Accuracy	z	Network architecture	Accuracy	z
00341	16-12-3	65.16%	1.738%	15-5-3	60.38%	0.869%
00293	17-1-3	70.63%	2.155%	3-12-3	76.13%	1.077%
00579	15-6-3	64.63%	2.285%	12-13-3	64.36%	1.142%
00010	17-4-3	69.69%	1.881%	10-2-3	68.64%	0.940%
00011	14-9-3	64.59%	1.738%	16-1-3	68.67%	0.869%
00044	7-3-3	65.65%	1.524%	13-2-3	63.23%	0.762%
00050	17-3-3	64.61%	1.660%	8-3-3	61.23%	0.830%
00005	14-18-3	61.76%	1.598%	9-3-3	60.92%	0.799%
00013	14-3-3	67.22%	2.007%	7-3-3	65.41%	1.003%
00016	17-1-3	69.08%	2.149%	10-1-3	68.15%	1.074%

Table 6.3 – Classification Result of Different Networks
with Different Decision Threshold z.

6.4.2 Experiment B

Genetic algorithm is applied to obtain the optimal network architecture, each chromosome represents a different network architecture. Two different decision thresholds have been tested, they are *avgMov* and *avgMov*/2. Since the average accuracy in the two models is nearly the same, it is found that different decision thresholds have no obvious effect to the classification performance.

Stock ID	Period – 1 Sept. 2000 to 1 Sept. 2001					
	Decision threshold $z = avgMov$			Decision threshold $z = avgMov/2$		
	Rested Budget	Stocks in Hand Values on 1 Sept. 2002	Total Gains	Rested Budget	Stocks in Hand Values on 1 Sept. 2002	Total Gains
00341	467,385	38,800	6,185	457,145	50,440	7,585
00293	490,375	8,850	-1,075	501,680	0	1,680
00579	497,720	2,610	330	498,050	2,610	660
00010	448,510	60,800	9,310	479,020	26,600	5,620
00011	545,540	43,875	89,415	421,685	131,625	53,310
00044	489,655	13,000	2,655	430,855	71,500	2,355
00050	484,550	15,750	300	501,350	0	1,350
00005	330,410	233,100	63,510	91,215	419,580	10,795
00013	561,235	0	61,235	502,845	68,750	71,595
00016	400,080	170,800	70,800	47,005	512,400	59,405
Total	4,715,460	587,585	302,665	3,930,850	1,283,505	214,355

Table 6.4 – Simulation Result for One-year Trading Days (1 Sept 2000 – 1 Sept. 2001)

Stock ID	Period – 1 Mar. 2001 to 1 Sept. 2001					
	Decision threshold $z = avgMov$			Decision threshold $z = avgMov/2$		
	Rested Budget	Stocks in Hand Values on 1 Sept. 2002	Total Gains	Rested Budget	Stocks in Hand Values on 1 Sept. 2002	Total Gains
00341	482,200	19,400	1,600	477,945	23,280	1,225
00293	487,525	8,550	-3,925	501,305	0	1,305
00579	498,260	2,030	290	497,850	2,610	460
00010	465,960	38,000	3,960	484,605	19,000	3,605
00011	475,055	43,875	18,930	391,645	131,625	23,270
00044	493,825	6,500	325	480,525	19,500	25
00050	499,650	0	-350	500,225	0	225

00005	299,255	233,100	32,355	90,940	491,580	10,520
00013	526,595	0	26,595	460,820	68,750	29,570
00016	354,690	170,800	25,490	25,875	478,240	4,115
Total	4,583,015	522,255	105,270	3,911,735	1,234,585	72,320

Table 6.5 – Simulation Result for Half-year Trading Days (1 Mar. 2001 – 1 Sept. 2001)

Stock ID	Period – 1 Jun. 2001 to 1 Sept. 2001					
	Decision threshold $z=avgMov$			Decision threshold $z=avgMov/2$		
	Rested Budget	Stocks in Hand Values on 1 Sept. 2002	Total Gains	Rested Budget	Stocks in Hand Values on 1 Sept. 2002	Total Gains
00341	492,335	7,760	95	477,165	23,280	445
00293	496,250	4,275	525	499,835	0	-165
00579	498,665	1,450	115	497,600	26,10	210
00010	481,925	19,000	925	489,625	11,400	1,025
00011	469,455	43,875	13,330	383,945	131,625	15,570
00044	493,125	6,500	-375	479,725	19,500	-775
00050	499,950	0	-50	500,025	0	25
00005	283,830	233,100	16,930	234,725	279,720	14,445
00013	507,670	0	7,670	440,990	68,750	9,740
00016	336,570	170,800	7,370	157,240	341,600	-1,160
Total	4,559,775	489,760	46,535	4,160,875	878,485	39,360

Table 6.6 – Simulation Result for Quarter-year Trading Days (1 Jun. 2001– 1 Sept. 2001)

As shown in Table 6.3, the number of hidden layers in all models is equal to one. Thus, one hidden layer is enough for most of models.

6.4.3 Experiment C

What investors are concerned most is how much advantage they can take in the stock market. Several simulation trading tests have been carried out. In each test, three typical sets of trading periods have been simulated. The first set of trading period is for one year, from 1 Sept. 2000 to 1 Sept. 2001 with totally 242 trading days. The second set of trading period is half a year, from 1 Mar. 2001 to 1 Sept. 2001 with totally 122 trading days. The last set of trading period is a quarter of year, from 1 June 2001 to 1 Sept 2001 with totally 61 trading days. Following procedures are used to decide whether to buy, sell or hold the stocks. Results are shown in Table 6.4 – 6.6. For each test, the investment budget *BG* is $500,000, it means that the total money in hand at the first trading day is $500,000. At the end of the trading period *endDay*, total gain is computed by:

Total gain = *BG* – (money in hand on *endDay* + stocks in hand values on *endDay)*

Step 1: Initialization

Total budgets *BG* is set to be $500,000.

The amount of shares in each 'Buy' or 'Sell' is 500.

Variable *S* indicated how many stocks in hand. The ratio of *S* to number of share is 1:500.

input(*t*) represents the input set in time *t*.

price(t) represents the stock price at time t.

Step 2: Use the set *input*(t) as the input of the network, derive the output.

Step 3: If Output = 1 and BG > 500**price*(t), buy the stock and set $S = S + 1$.

BG = BG – 500**price*(t)

Step 4: Else If Output = -1 and $S > 0$, sell the stock.

Set $S = S - 1$.

BG = BG + 500**price*(t).

Step 5: Else, Hold

Step 6: If BG <=0 and S<=0, stop

Step 7: If t > total number of trading days (out of simulation period), end the test.

Else, set $t = t + 1$ and return to Step 2.

As shown in Tables 6.4 – 6.6, it is found that the use of a greater decision threshold can generate greater profits among the 10 stocks. However, a greater gain means greater risk, the network having a greater decision threshold may lead to a greater chance of loss.

This agent makes use of a genetic neural network for addressing the major problem when using artificial neural networks in financial forecasting problems. It includes the input selection, network architecture and the output format.

A *decision threshold* is also introduced to help define the best strategy for investors. It is found that common technical indictors such as RSI and Moving Average can be ignored in the network input selection. Also, preprocessing of the raw data is necessary for a good network.

We transformed the forecasting problem into a classification problem by using a decision threshold. This agent gives out suggestion on one of the three investment strategies: buy, sell or hold. It is found that using greater decision threshold, investors could have greater risk and return.

Chapter 7 Conclusion and Future works

7.1 Conclusion

In this research project, we proposed a multi-agent system in solving the non-linear financial forecasting problem. In the last two decades, people attempted to use different AI technologies (neural network, expert system, fuzzy logic, etc.) to solve the problem. It turns out that the solutions are often not accurate enough, lack of explanatory power or dependent on some domain experts. The proposed system consists of five forecasting agents and two non-forecasting agents. The five forecasting agents are Fundamental Agent, Technical Agent, Associate Agent, Adaptive Agent and Expert Agent. Different agents give different recommendations in the investment process. And the two non-forecasting agents are Information Gathering Agent and Coordinate Agent. The Information Gathering Agent collects all the useful information for different forecasting agents. The Coordinate Agent collects all the recommended strategies from different forecasting agents, leading to a final recommended strategy for the users.

Each agent has different mechanism and embedded with different complicated algorithms in it. Completing all the agents in the proposed system is time consuming and need huge efforts to do so. As such, this study just focuses on the construction of the three agents the Coordinate Agent, Technical Agent and Adaptive Agent. These three agents are the most complicated and representative agents in the proposed system.

The motivation of doing this research is targeting the development of a system which incorporates most advantages of different AI technologies and methods in solving the non-linear financial forecasting problem. In literature, the approach has in at least two directions, one is based on inferring rules from past and current behaviour of market data, it is led by some inductive and learning-based technique such as neural networks [2] or fuzzy logic [13]. Another direction uses physical and mathematical models [18] based on different economic prototypes. It attempts to find dynamic indicators derived from physical models based on general principles of nonequilibrium stochastic process [31] that reflect certain market factors. It seems that most of the developments in literature always give rise to bias and contradictory results based on somewhat different approaches of analysis. All the literature developments have their own advantages and disadvantages. It is hardly any one of it which can give an accurate, understandable and non-bias prediction.

In order to solve the problem, we introduced a multi-agents system with different agents using different approaches to address the problems. The proposed system incorporates the advantages from different approaches and leads to a final recommendation for the users. This is done by the Coordinate Agent discussed in chapter 4.

Chapter 4 illustrates how to collect different suggestions given out by different forecasting agents respectively and how to incorporate it them to give a better solution. In decision theory [36], most popular mechanism in solving such problem is the simple majority rule [36] but found to be insufficient to solve our problem. It is mainly due to the intransitive group reference [36]. In our case, we have introduced the *prefer ratio, score tables* together with two weighting methods: *simple weighting* and *exponential weighting*. Extensive experiments have been carried out and proved that the *prefer ratio* with the *exponential weighting* method can achieve our goals and have better overall system performances. We then examine these agents and approaches in the following.

Technical Agent is based on the technical analysis and use Case-based Reasoning method to forecast the stock market movements. However, technical analysis mainly focuses on analyzing the chart patterns, which is a non-trivial task. Because one time scale alone cannot be applied to all analytical processes [2], the identification of typical patterns on a stock price chart is dynamic and non-trivial. To solve this problem, we have introduced an algorithm PXtract which dynamically makes use of different time windows to identify possible chart patterns. For the recognition and identification process in algorithm PXtract, the wavelet multi-resolution analysis and radial basis function neural network (RBFNN) matching are applied. In the training process of the RBFNN, the number of training sets is an important factor for a well training of the network. Since extracting all the chart pattern template in time series data is infeasible, expensive and time consuming, we have introduced the *radial deformation* method to generate more training examples for the network. With our approach, the average recognition rate of the proposed method has reached 81%.

Adaptive Agent uses a typical Multi-layer Feed Forward Neural Network to do forecasting. Input selections and topology of the network architecture is the major issues of building a good network model. We have introduced a generic neural network for addressing the problems. Moreover, a *decision threshold* is created to help define the best strategy for decision making by investors. It is found that common technical indictors such as RSI and Moving Average can be ignored in the network input selection. Also, preprocessing of the raw data is necessary for a good network. In this agent, we transformed the forecasting problem into a classification problem by using the *decision threshold.* The agent gives out suggestion on one of the three investment strategies: buy, sell or hold. It is found that using greater *decision threshold*, greater risk and return the investors have.

Overall, we have contributed to the following aspects including the development of the multi-agents system, three agents including the Coordinate Agent, the Technical Agent and the Adaptive Agent. In the Coordinate Agent, we introduced a mechanism to collect all the recommendations from different agents and synthesize them into one recommendation. In the Technical Agent, we introduced an algorithm PXtract which makes use of different time scale windows to identify possible chart patterns. In the Adaptive Agent, a generic approach is introduced for the decision of the network architecture of neural network and the Input Selection process.

7.2 Future Works

7.2.1 Large Scale Chart Patterns Database

Currently, we have only 14 different chart patterns templates with totally 308 training samples. According to Thomas[38], there has 47 different chart patterns which can be extracted out from the time series data. To complete the CBR system in the Technical Agent, building templates of all the remaining different chart patterns is required.

7.2.2 Speeding Up the Genetic Processes in the Adaptive Agent

Since genetic algorithm is the base engine within the Input Selection and Network Architecture Decision Process, the workload of individuals within the population can be evenly distributed over different CPUs with little additional efforts. In most cases, the time required for searching the solutions can be dramatically reduced depending on the population size.

7.2.3 Fundamental Agent

Ongoing work on the iWAF includes building the rest of the three forecasting agents with including the Fundamental Agent, Associate Agent and Expert Agent. Fundamental agent performs fundamental analysis using information such as P/E ratio, book values in the case of stocks, crop reports or import/export figures in the case of commodity futures, a company's finance, forecasts of sales, earnings, and expansion plans.

7.2.4 Associate Agent

Associate agent investigates the dynamics of human factors and psychological impacts. This can be done by extending the functionality of the Technical Agent.

Technical Agent uses the Algorithm PXtract to extract all possible hidden chart patterns from the time series data. We can further study the Technical Agent by applying some data mining analyses to examine the psychological behaviors of investors and their reactions and impacts during the movements of stock prices.

7.2.5 Expert Agent

Expert agent processes AI heuristics for the model of trading rules, risk and return calculation, and investment strategies acting on indicators derived from the market. It also provides personalized advice and substantiation of recommendations.

Different means of optimizing the performance of these agents are useful. We anticipate that the system can provide different measures of the financial market and allow investors to position their investment better and more effective.

References

[1] Ahmed Riahi-Belkaoui, *Financial Analysis and the Predictability of Important Economic Events*, (Greenwood Publishing, pp. 8 – 14, pp.34 – 38, 1998)

[2] Austin, M., Looney, C. and Zhuo, J., *Security market timing using neural networks*, (The New Review of Applied Expert System, 1997)

[3] Boris K and Evgenii, V. *Data Mining in Finance – Advance in Relational and Hybird Methods*, KluWer Academic, 2000

[4] Christodoulou Christos, *Applications of Neural Networks in Electromagnetics* (Boston, Mass: Artech House, 2001)

[5] Corby, O. and Dieng, R. (1999), "The WebCokace Knowledge Server", IEEE Internet Computing, 3(6), pp. 38-43.

[6] Dan W. Patterson, *Neural Networks Theory and Applications* (Prentice Hall, 1996)

[7] Danny B.Lange, Mitsuru Oshima, *Programming and Depolying JAVA Mobile Agents with Aglets*, (Addison Wesley, 1998)

[8] Daubechies, I. "Orthonormal Bases of Compactly Supported Wavelets", *Comm. Pure Appl. Math.* 41, 909-996, 1988.

[9] Edward Gately, *Neural Networks for Financial Forecasting – Top techniques for Desingning and Applying the Latest Trading Systems*, Wiley Trader's Advantage, 1996

[10] Fahlman, Scott E., Lebiere, Christian, "The Cascade-Correlation Learing Architecture", *Technical Report CMU-CS-90-100,* School of Computer Sicence, Carnegie Mellon University, Pittsburgh, PA 1991.

[11] Feng, L, Dillon, T. and Liu, J. (2001), "Inter-transactional association rules for prediction and their application to studying meteorological data", conditional accept by *Data and Knowledge Engineering*, Elsevier.

[12] Firth, Michael, *The value of Shares and the efficient-markets theory*, London, Macmillan, 1997

[13] Funabashi, M., Maeda, A., Morooka, Y. and Mori, K AI in Japan, Fuzzy and Neural Hybird Expert Systems, Synergetic AI, *IEEE Expert*, pp. 32-40, 1995

[14] Fu-lai Chung, Tak-chung Fu, Luk, R., Ng, V., *Evolutionary time series*

segmentation for stock data mining, in Proceeding of 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan, 2002, 83-90

[15] Fu-lai Chung, Tak-chung Fu, Robert W.P. Luk, Vincent Ng, *Flexible Time Series Pattern Matching Based on Perceptually Important Points,* in IJCAI-01 Workshop on Learning from Temporal and Spatial Data, Seattle, USA, 1-7, 2001.

[16] http://web3.asia1.com.sg/timesnet/data/hk97/intro.html

[17] http://www.mathworks.com

[18] Iman, R.L. and Conover, W.J, *modern Business Statistics*, (Wiley, 1989)

[19] Joseph P.B. and Jennifer B. (2001), "Constructing Intelligent Agent using Java", Wiley

[20] K Kamijo and Tanigawa, T. (1990), "Stock price pattern recognition: a recurrent neural network approach", in *Proceedings of IJCNN'90*, 1, pp. 215-221.

[21] Kamijo, K, Tanigawa, T., *Stock Price Pattern Recognition – A Recurrent Neural Network Approach,* in 1990 International Joint Conference on Neural Network, San Diego, 1990, 215-222

[22] Lee, R.S.T. and Liu, J.N.K. (2000a), "An approach using Elastic Graph Dynamic Link model for automating the satellite interpretation of tropical cyclone patterns", In *Invariants for Pattern Recognition and Classification* (Series in Machine Perception and Artificial Intelligence 42), pp. 189-207, editor M. A. Rodrigues, World Scientific

[23] Lee, R.S.T. and Liu, J.N.K. (2000b), "Tropical Cyclone Identification and Tracking System using integrated Neural Oscillatory Elastic Graph Matching and hybrid RBF Network Track mining techniques", *IEEE Transaction on Neural Network*, **11**(3), pp. 680-689.

[24] Lee, R.S.T. and Liu, J.N.K. (2000c), "A fuzzy network based shopping agent in E-commerce environment", in *Proceedings of International ICSC Symposium on Multi-Agents and Mobile Agents in virtual organizations and E-commerce (MAMA'2000),* December 11-13, 2000, Wollongong, Australia

[25] Liu, James N.K. and Jane You (2000), "An Agent-based intelligent system for E-commerce applications", in *Proceedings of International ICSC Symposium on Multi-Agents and Mobile Agents in virtual organizations and E-commerce (MAMA'2000),* December 11-13, 2000, Wollongong, Australia.

[26] Liu, James N.K. and Leung, Tommy T.S., A Web-based CBR agent for

financial forecasting, *Workshop Program at the 4th International Conference on Case-Based Reasoning 2001*, 30 July – 02 August, Vancouver, Canada, pp. 243-253, 2001

[27] Liu, J.N.K. and Fung, G.S.K, *Signature verification based on Fuzzy Genetic Algorithm*, (Jain and Lazzerini, Eds., Intelligent Techniques in Character Recognition: Practical Applications. CRC Press, pp. 121-148,1999)

[28] Liu, J.N.K. and Sin, K.Y., Fuzzy neural networks for machine maintenance in Mass Transit Railway System, *IEEE Transactions on Neural Networks*, Vol.8, 4, pp. 932-941, 1997

[29] Liu, N.K. (2001), "An intelligent business advisory system for stock investment", *Encyclopedia of Microcomputers.*, Vol. 27, pp. 161-184.

[30] M. Klusch (eds) (1999), "Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet", Springer-Verlag Berlin Heidelberg.

[31] Makridakis, S., Wheelwright, S.C. and Hyndman, R.J., *Forecasting Methods and Application*, 3rd ed, (New York, Wiley, 1998)

[32] Mallat, S. Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Trans. Amer. Math. Soc.*, 1989, 69-87.

[33] Park, J., and I.W. Sandberg, Approximation and Radial Basis Function Networks, *Neural Compuation, (vol. 5)*, 1993, 305-316.

[34] Park, J., and I.W. Sandberg, Universal Approximation Using Radial Basis Function Networks, *Neural Computation, (vol.3)* 1991, 246-257.

[35] Rick Bensignor, *New Thinking in Technical Analysis*, (Bloomberg Press, 2000)

[36] Simon French, *Decision Theory - An Introduction to the Mathematics of Rationality*, (Ellis Horwood Limited, 1986)

[37] T Plummer, T., *Forecasting Financial Markets* (Kogan Page Ltd, 1993)

[38] Thomas N.B., *Encyclopedia of Chart Patterns* (John Wiley & Sons, 2000)

[39] Tou, J. T, and R. C. Gonzalez, *Pattern Recognition Principles, Reading*, (MA: Addison Wesley 1974).

[40] Vollrath, I., Wilke, W., Bergmann R.: Intelligent Electronic Catalogs for Sales Support – Introducing Case-Based Reasoning Techniques to On-Line Prodct Selection Applications. In Roy, R., Furuhashi, T., Chawddhry, P.K. (Eds.): *Advances in Soft Computing – Engineering Design and Manufacturing*. Springer. London, 1999

[41] Weinstein, S., *Stan Weinstein's secrets for profiting in bull and bear*

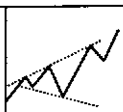
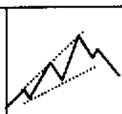
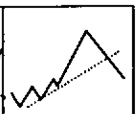
markets, (McGraw Hill, 1988)

[42] Widrow, B., R.G. Winter, R.A. Baxter, "Learning Phenomena in Layered Neural Networks". *1st IEEE international Conference on Neural Networks*, San Diego, USA, 1987

[43] W.R. Uttal, T. Baruch, L. Allen, "The Effect of Combinations of Image Degradations in a Discrimination Task", Perception & Psychophysics, vlo.57, no. 5, pp.668-681, 1995.

[44] Y.Y.Tang, L.H. Yang, J.Liu, H. Ma, *Wavelet Theory and Its Application to Pattern Recognition* (River Edge, NJ: World Scientific Publsihing, 2000)

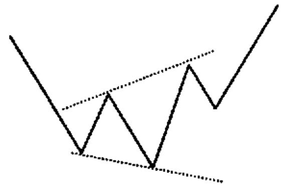
Appendix A

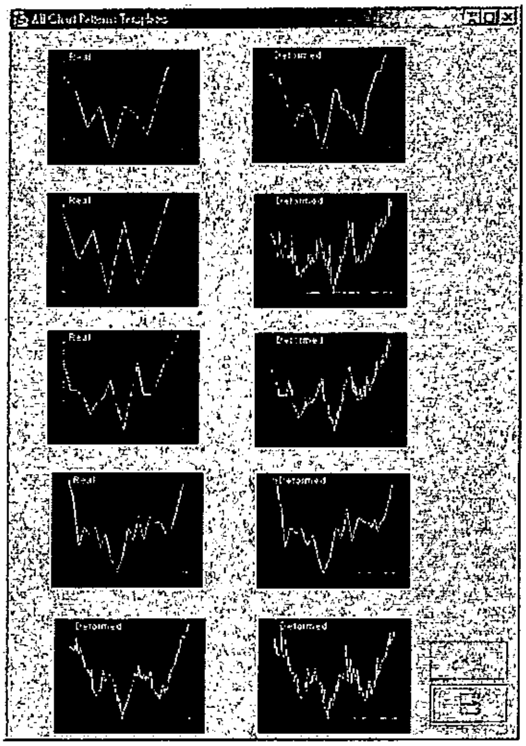
	 Broadening Bottoms	 Broadening Formations, Right-Angled and Ascending	 Broadening Formations, Right-Angled and Descending	 Broadening Tops	 Broadening Wedges, Ascending	 Bump-and-Run Reversal Bottoms	 Bump-and-Run Reversal Tops
Stock ID	00016	00293	00011	00005	00016	00016	00016
From	18/5/1999	7/6/99	8/8/00	1/2/95	26/3/99	21/9/00	3/7/97
To	2/6/1999	22/7/99	26/10/00	14/4/95	20/5/99	13/12/00	20/8/97
Stock ID	00016	00293	00005	00005	00016	00016	00016
From	8/8/99	28/11/98	13/11/97	28/5/96	5/8/00	26/7/97	4/11/96
To	20/8/99	12/2/99	27/2/98	24/7/96	16/9/00	16/9/97	13/12/96
Stock ID	00293	00293	00005		00341	00016	00341
From	13/9/99	29/5/98	28/10/96		10/5/98	28/6/95	1/9/98
To	8/10/99	14/8/98	12/1/97		3/6/98	31/8/95	9/10/98
Stock ID	00011	00293			00341	0341	
From	3/7/96	28/1/97			6/8/99	5/2/01	
To	5/8/96	213/97			16/9/99	2/3/01	
Stock ID		00011			00011	00341	
From		25/7/01			27/10/99	3/2/95	
To		7/9/01			6/1/00	16/3/95	
Stock ID		00005			00005	00293	
From		12/11/99			20/2/98	18/2/97	
To		13/11/00			14/5/98	9/4/97	
Stock ID						00293	
From						19/9/95	
To						7/12/95	
Stock ID						00011	
From						20/3/97	
To						1/12/98	

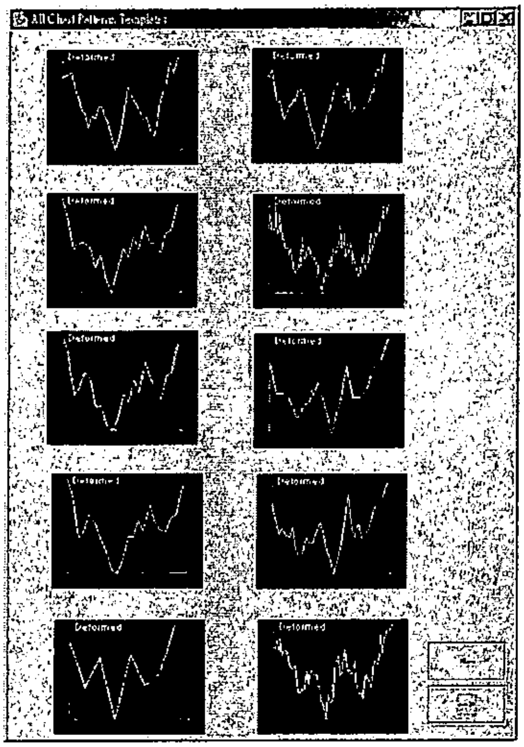
	 Cup with Handle	 Double Bottoms	 Double Tops	 Head-and-Shoulders, Top	 Head-and-Shoulders, Bottoms	 Triangles, Ascending	 Triangles, Descending
Stock ID	00016	00016	00016	00016	00016	00016	00016
From	5/7/00	16/2/00	7/8/00	8/9/95	6/8/99	16/12/95	7/8/97
To	25/7/00	6/3/00	28/10/00	29/10/95	20/8/99	4/2/96	2/9/97
Stock ID	00016	00016	00341	00016	00016	00293	00016
From	25/7/95	23/11/98	15/6/98	13/4/99	17/5/99	1/3/00	12/10/97
To	15/9/95	19/12/98	2/9/98	21/5/99	2/6/99	7/4/00	12/11/97
Stock ID	00341	00016	00011	00341	00293		00011
From	2/10/00	5/7/96	12/8/99	26/10/01	12/9/00		30/12/99
To	27/10/00	9/8/96	28/9/99	29/11/01	21/11/00		11/2/00
Stock ID	00341	00016	00005	00011	00005		
From	24/6/99	4/1/95	9/6/99	15/11/99	9/12/97		
To	17/799	30/1/95	22/7/99	24/12/99	19/2/98		
Stock ID		00341	00005	00011			
From		5/11/99	28/11/95	17/12/96			
To		4/1/00	2012/95	16/2/97			
Stock ID		00293					
From		8/12/97					
To		28/1/98					
Stock ID		00293					
From		20/7/98					
To		21/8/98					
Stock ID		00011					
From		26/8/96					
To		1/10/96					
Stock ID		00011					
From		28/3/95					
To		11/5/95					
Stock ID		00005					
From		13/7/99					
To		25/8/99					

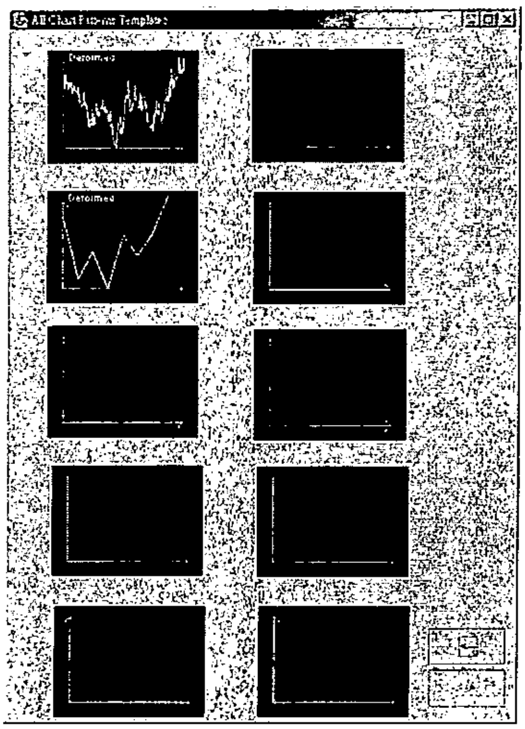
Appendix B

Training Set for Broadening Bottoms

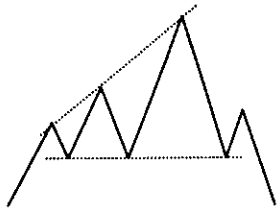


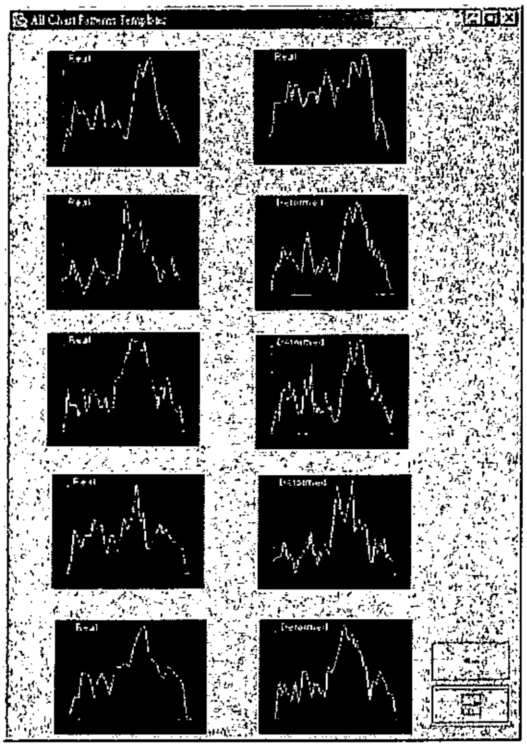


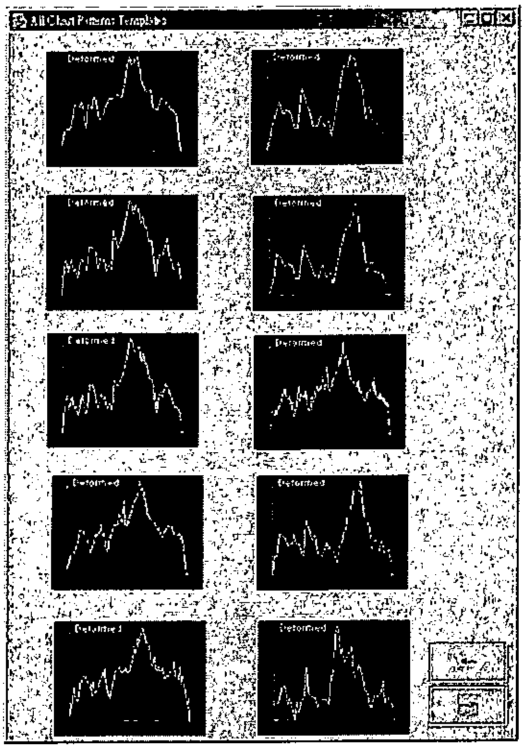


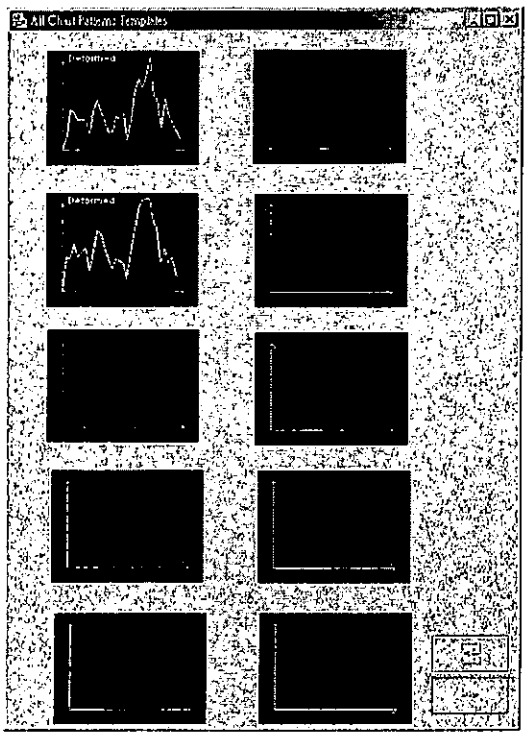


Training Set for Broadening Formations, Right-Angled and Ascending

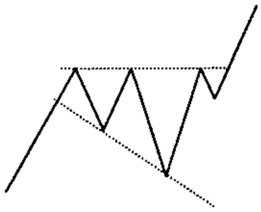


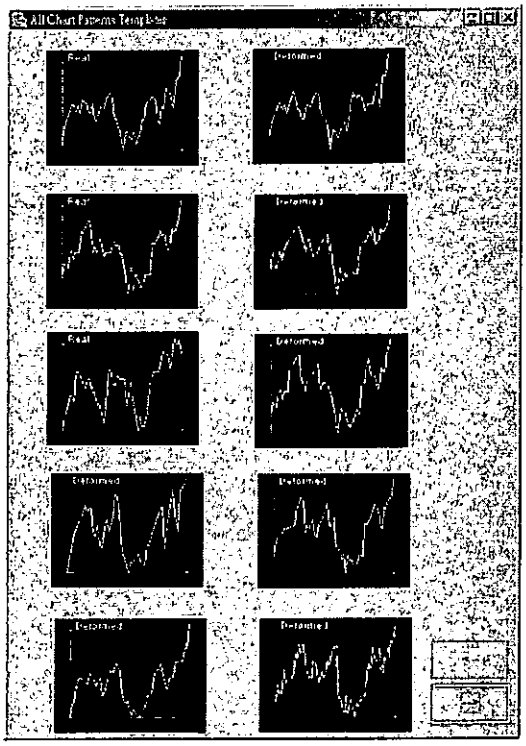


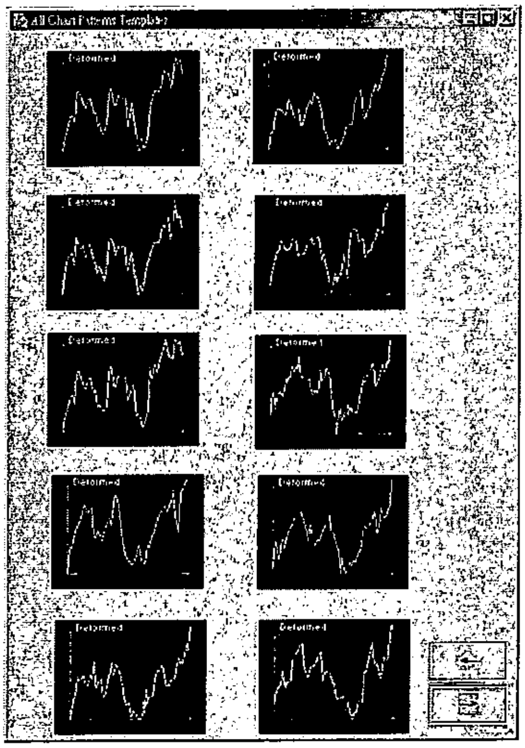


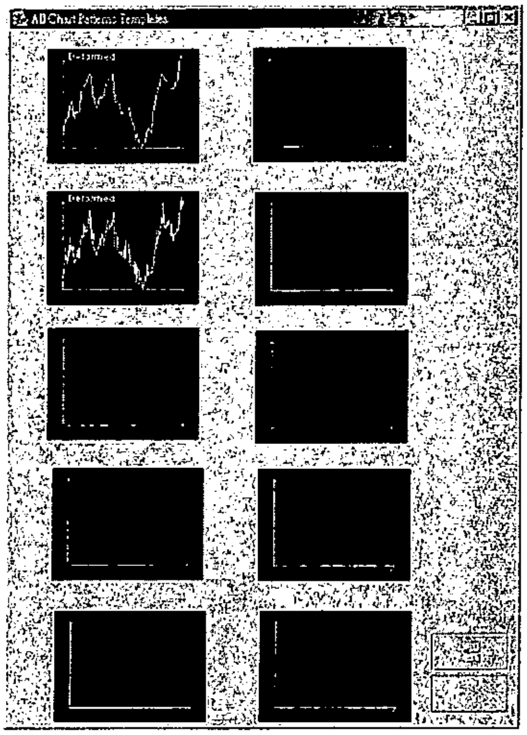


Training Set for Broadening Formations, Right-Angled and Descending

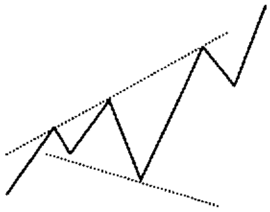


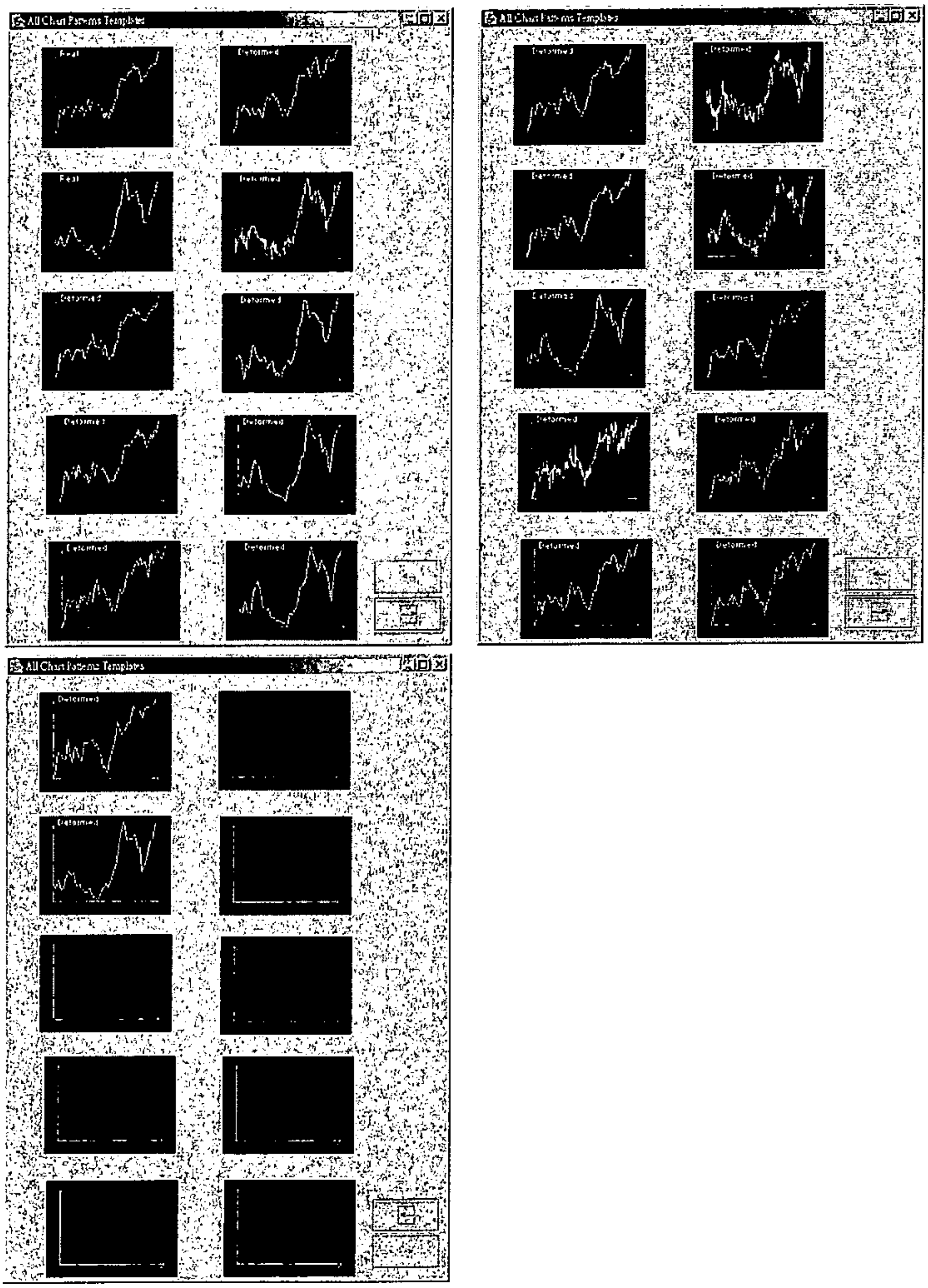




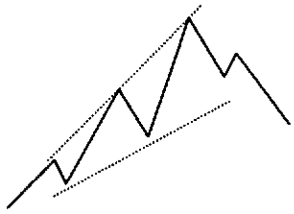


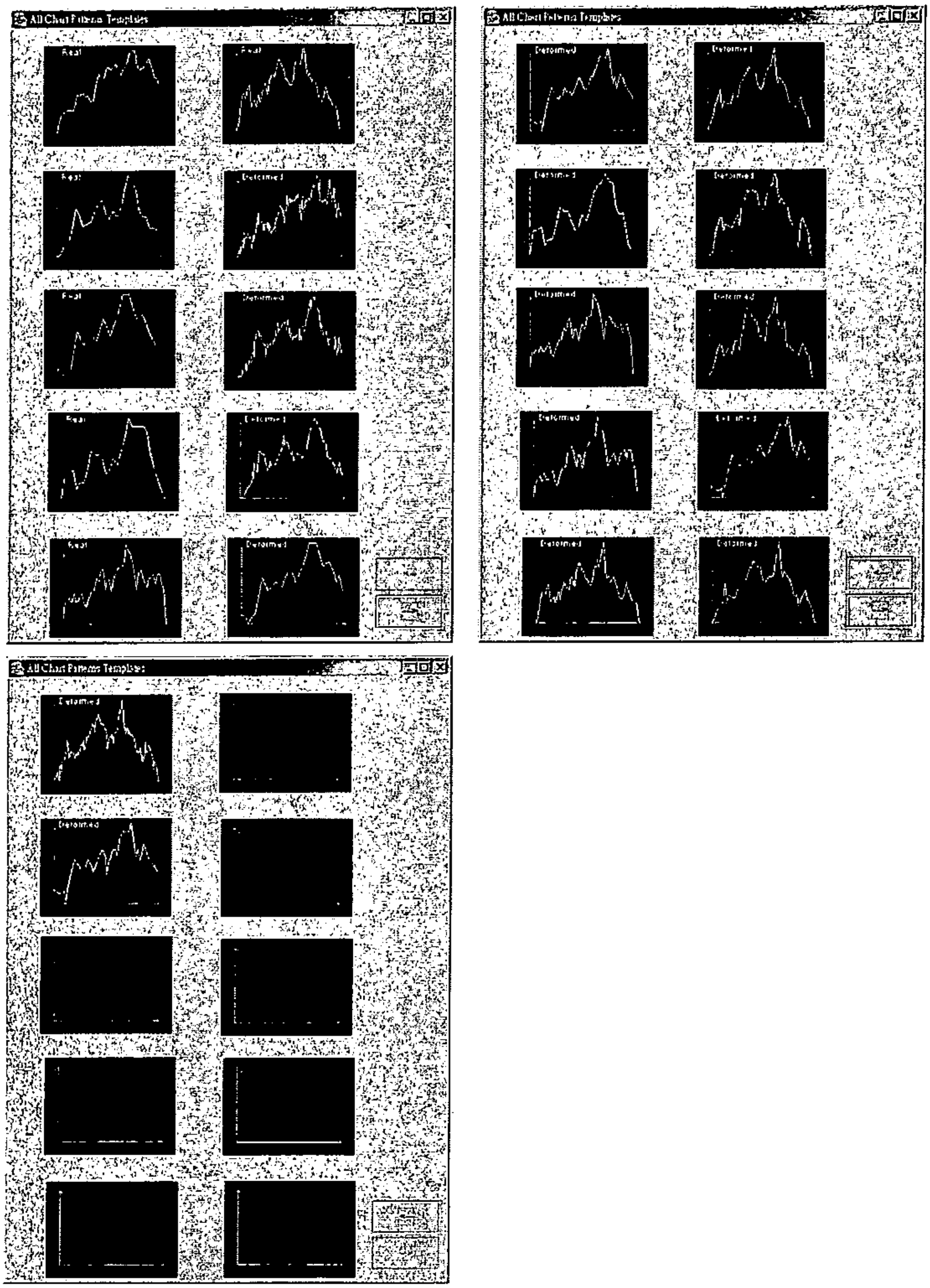
Training Set for Broadening Tops



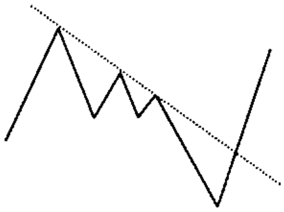


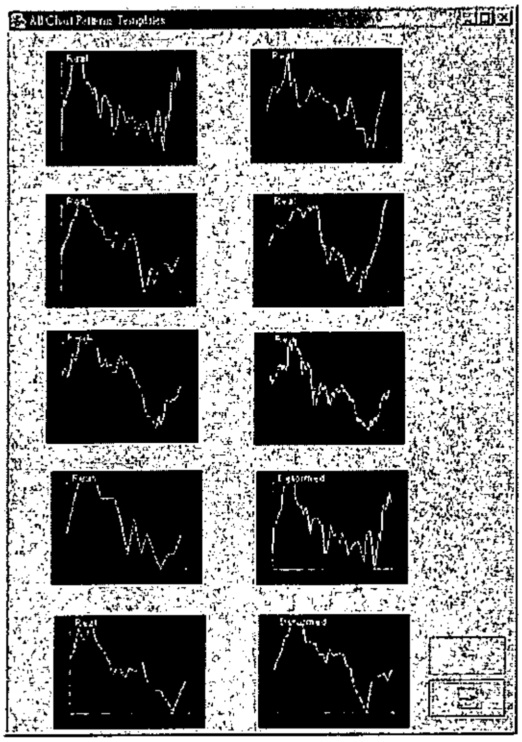
Training Set for Broadening Wedges, Ascending

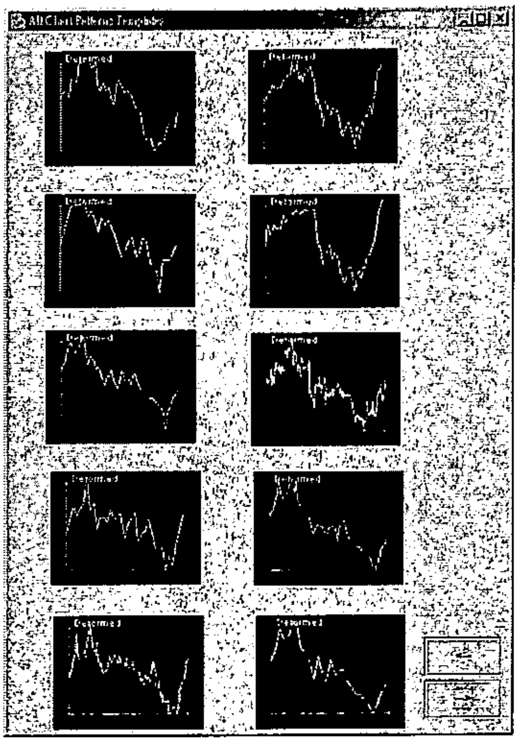


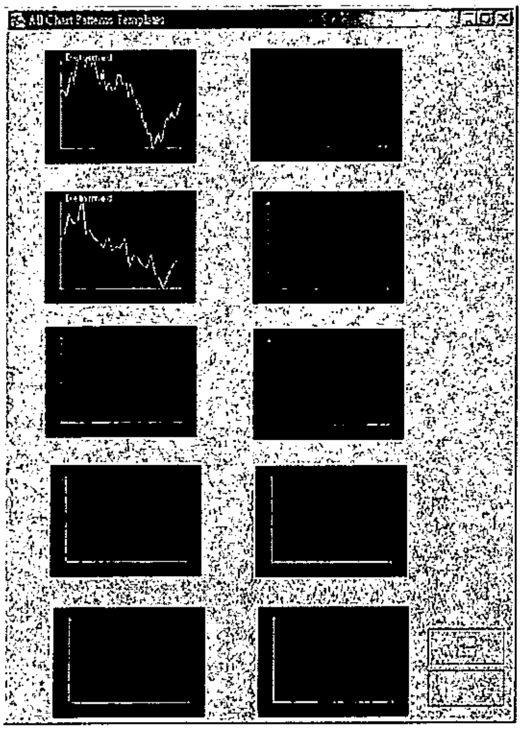


Training Set for Bump-and-Run, Reversal Tops

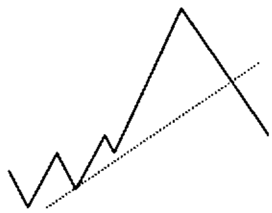


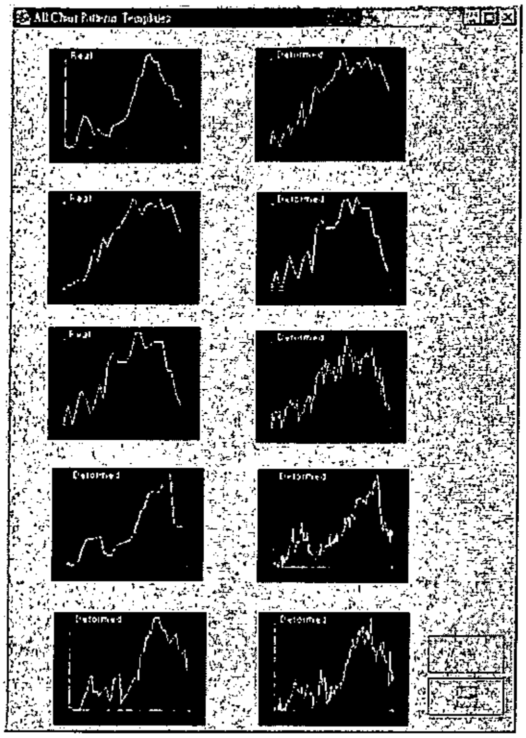


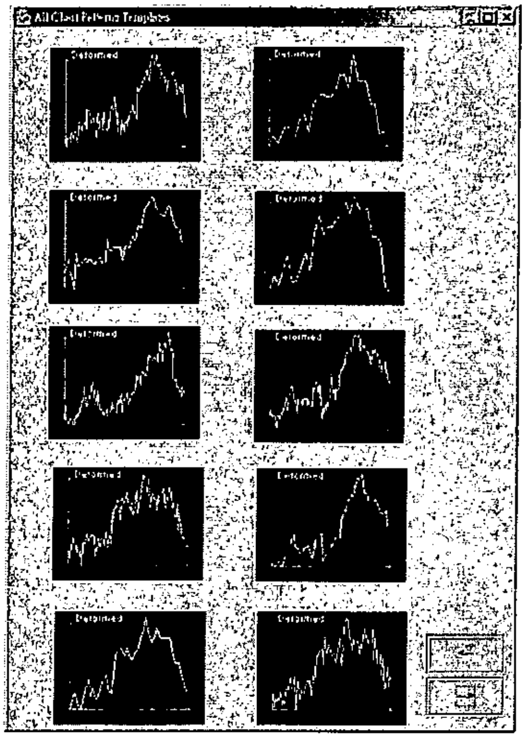


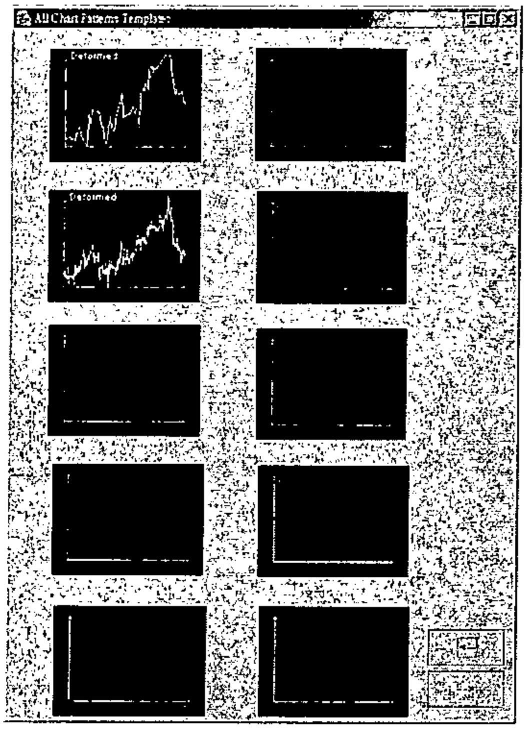


Training Set for Bump-and-Run, Reversal Tops

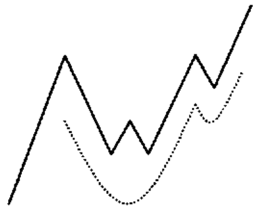


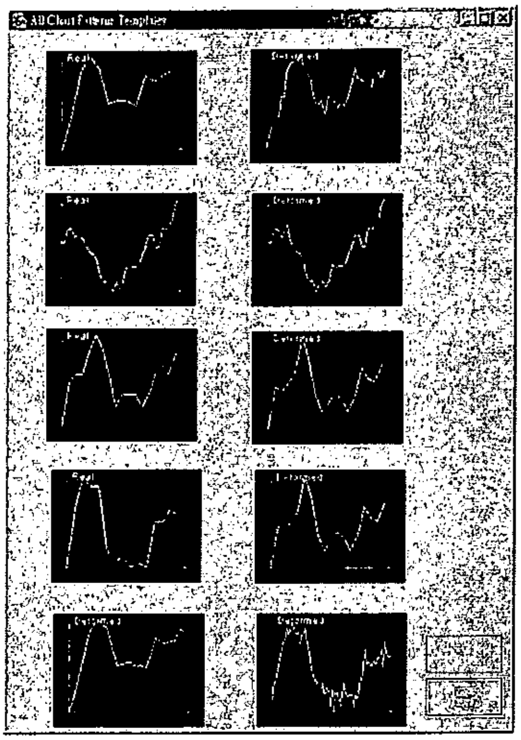


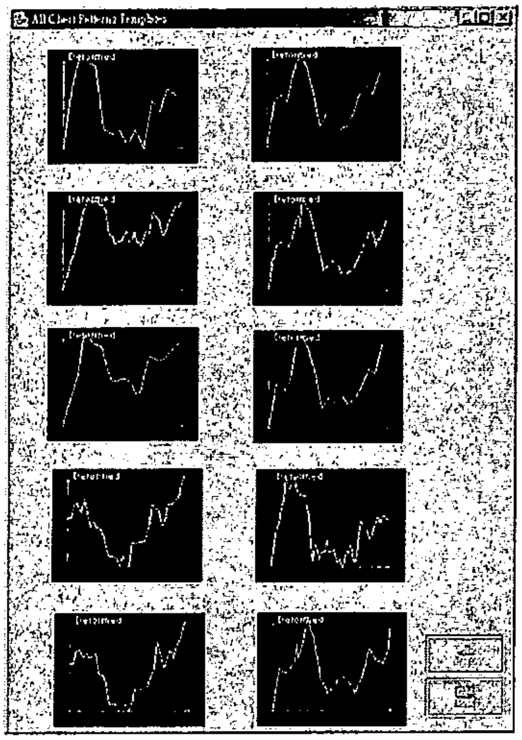


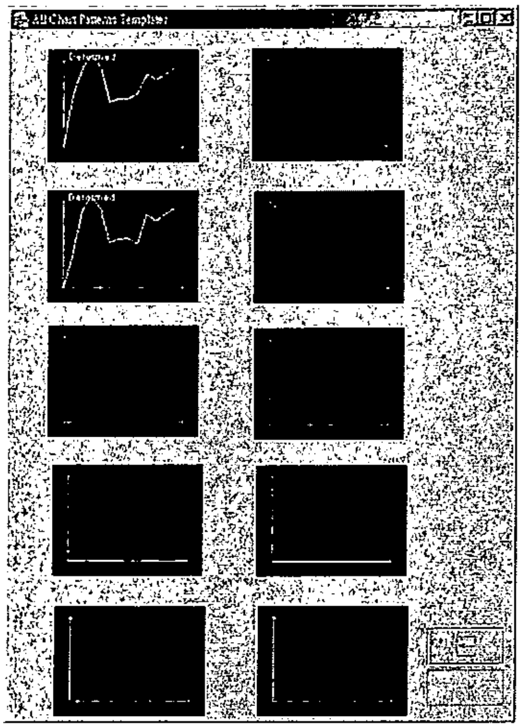


Training Set for Cup with Handle

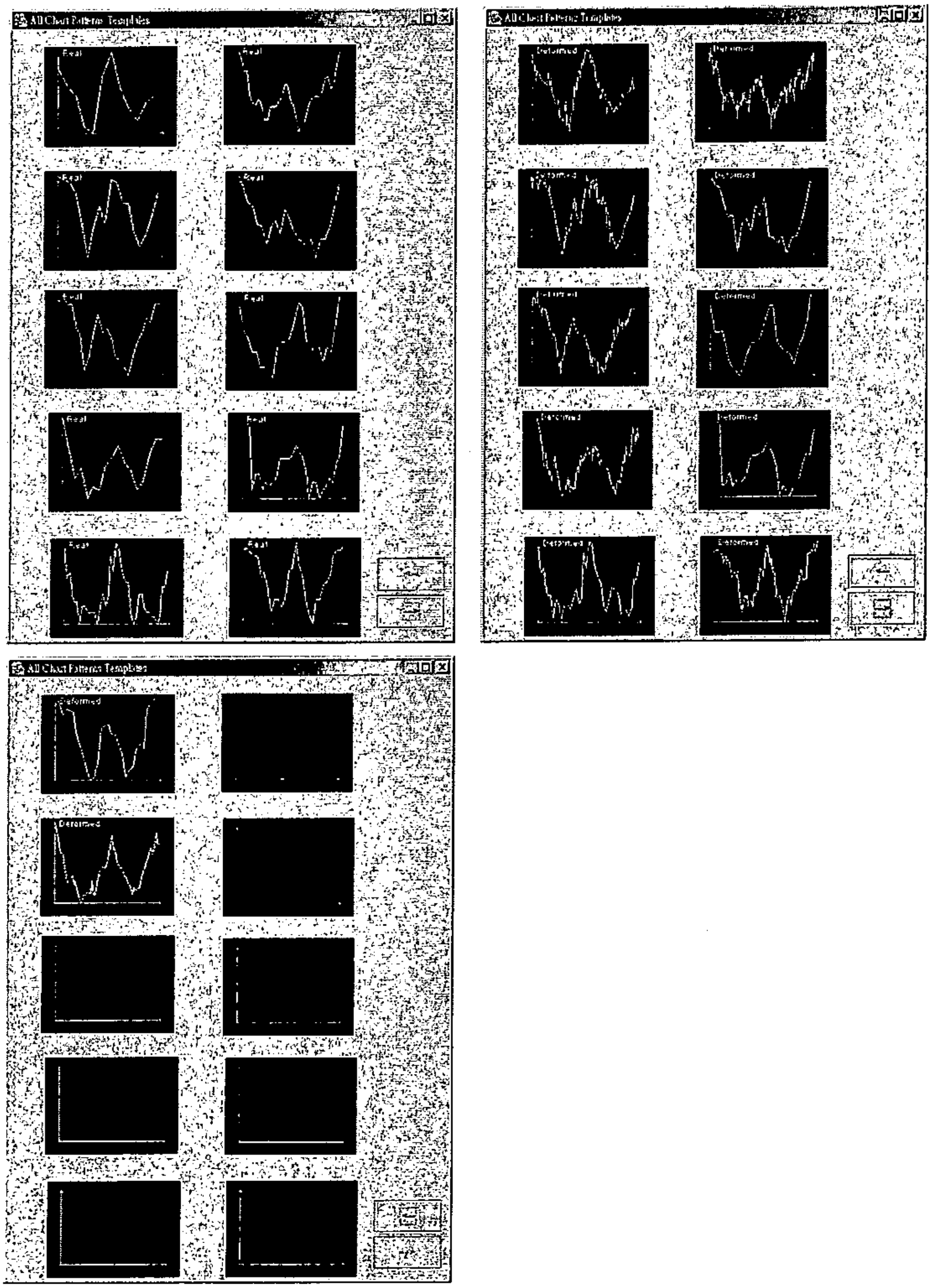




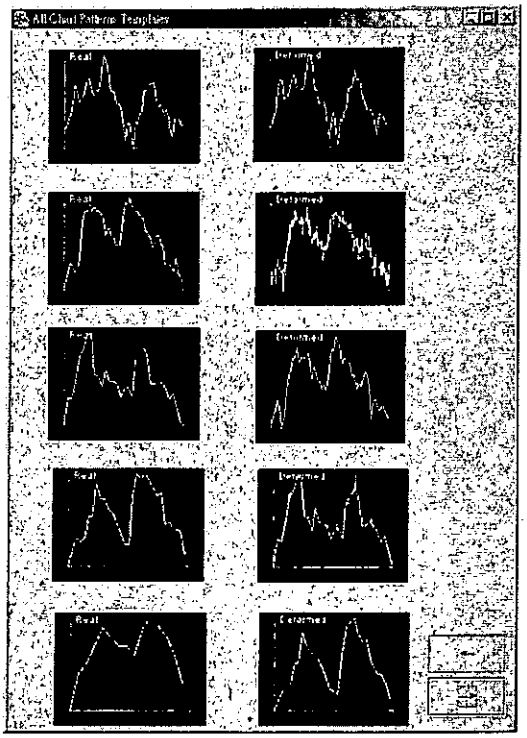


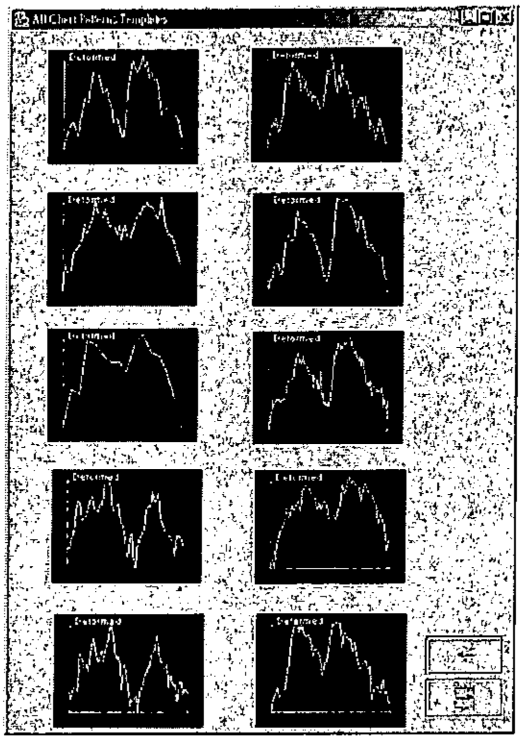


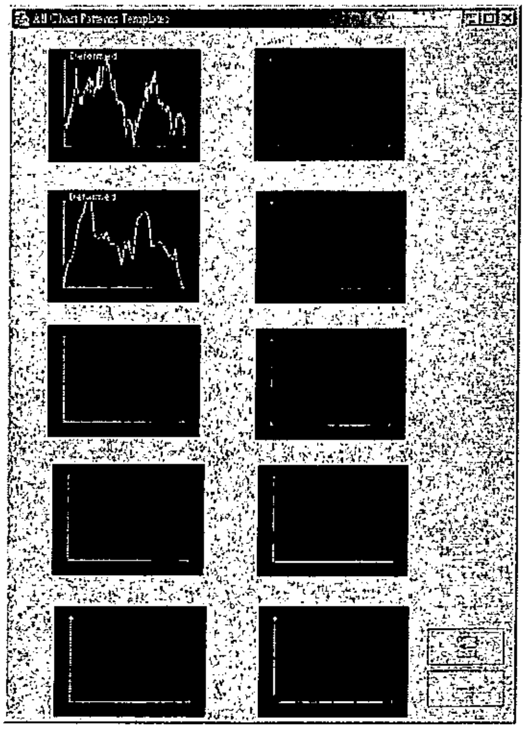
Training Set for Double Bottoms



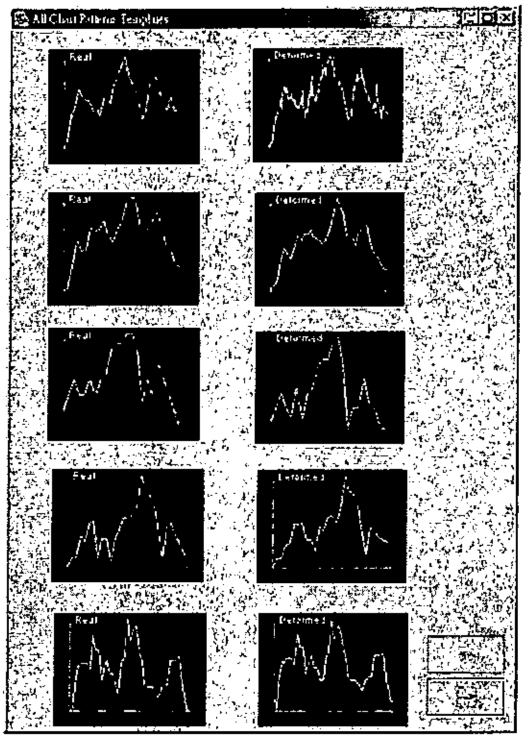
Training Set for Double Tops

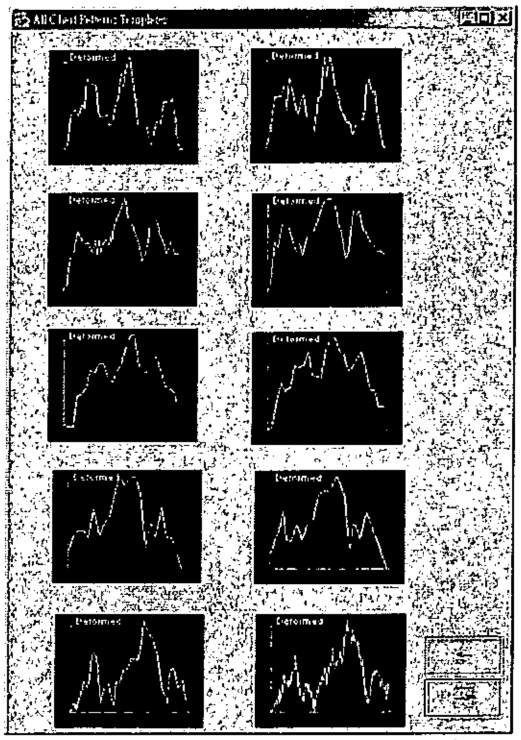


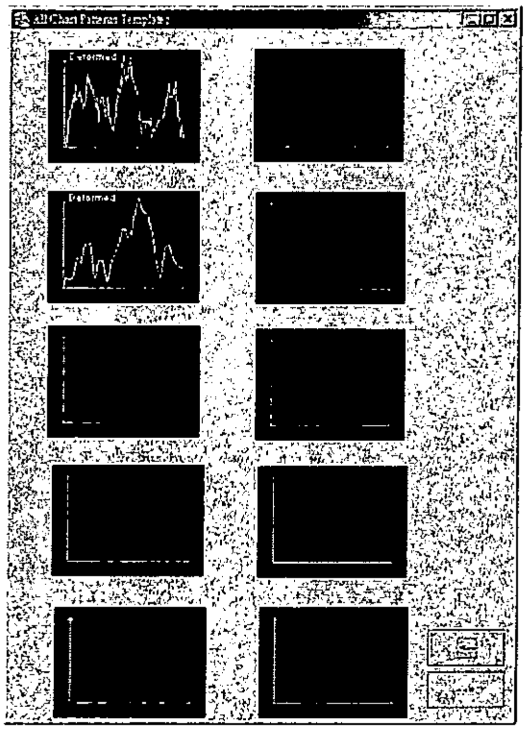




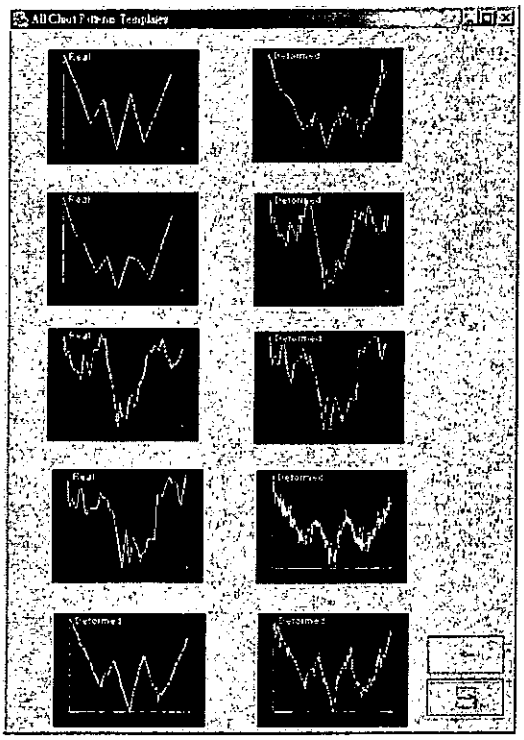
Training Set for Head-and-Shoulders, Top

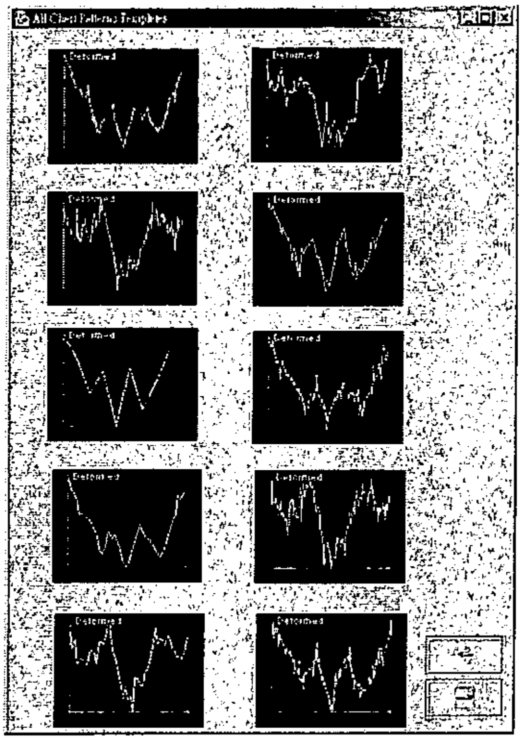


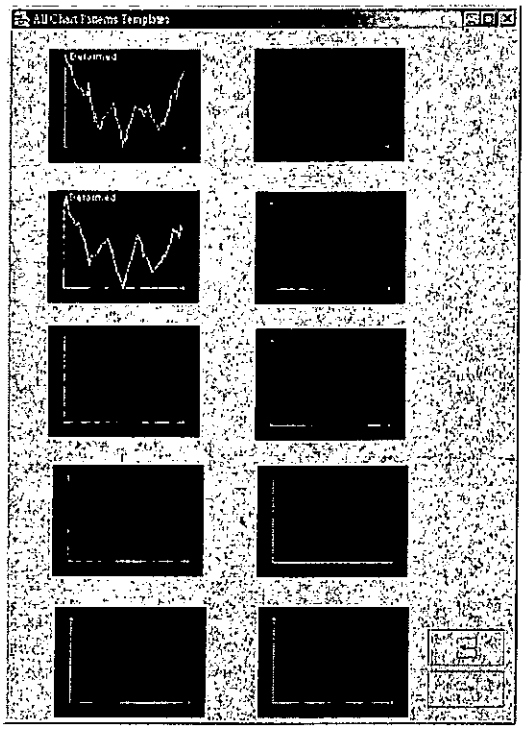




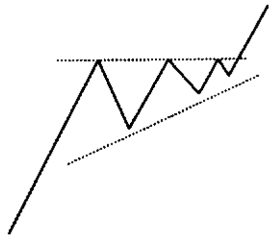
Training Set for Head-and-Shoulders, Bottoms

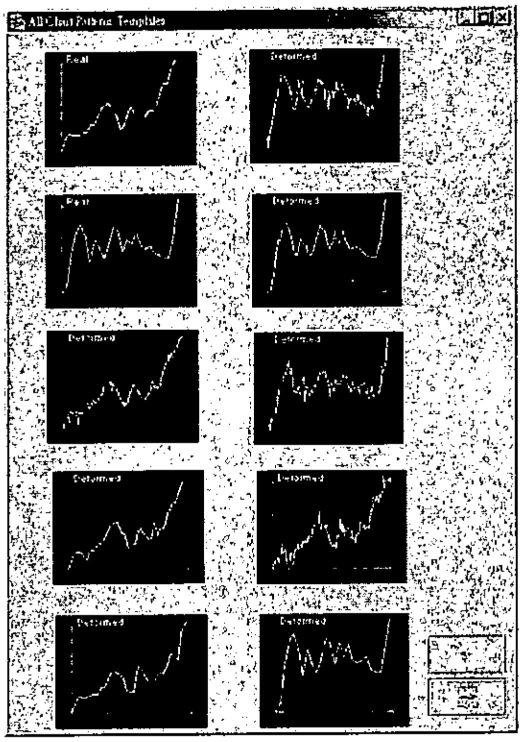


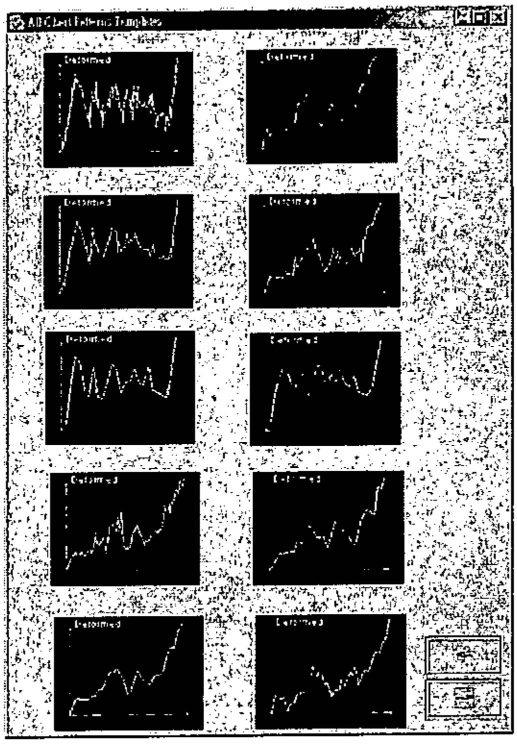


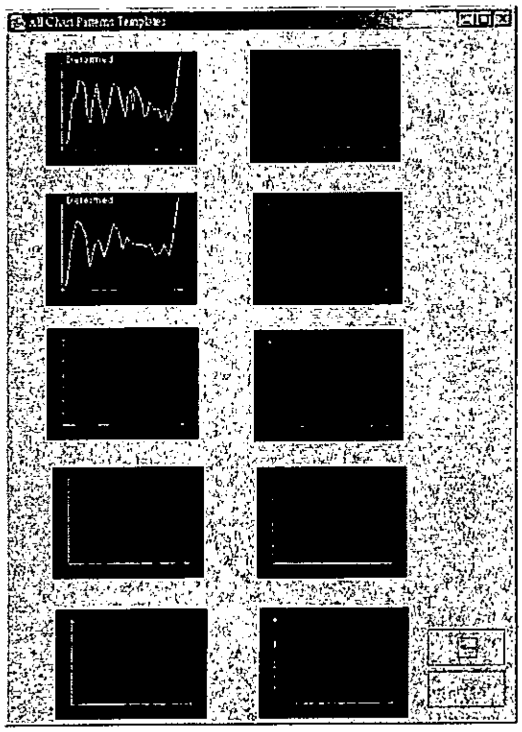


Training Set for Triangles, Ascending

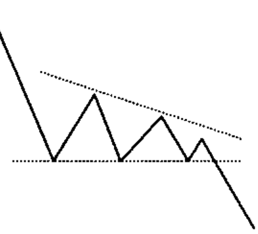


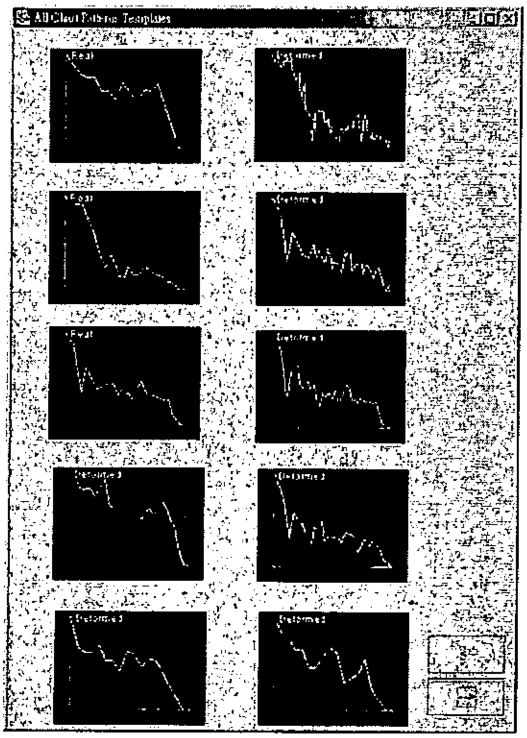


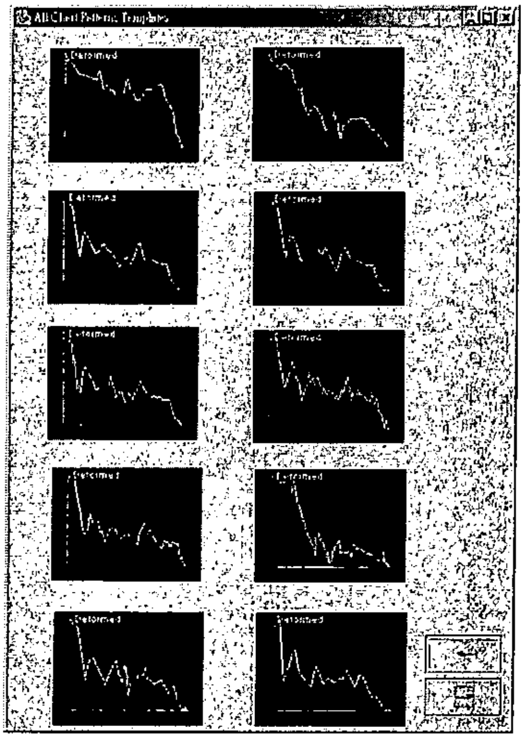


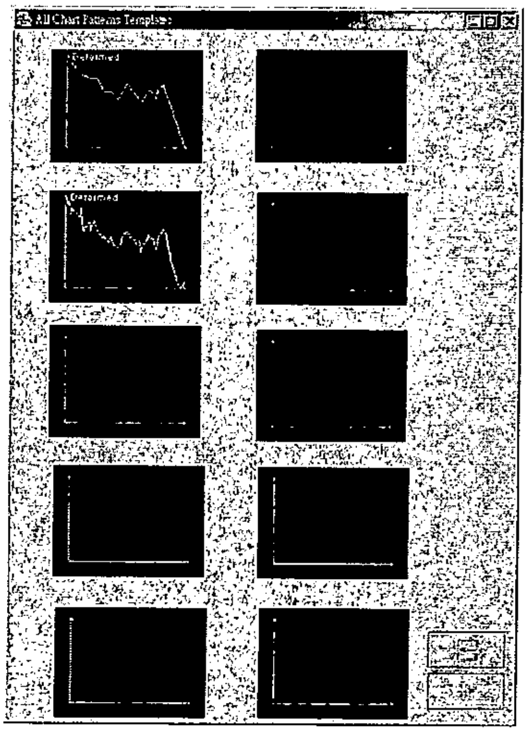


Training Set for Triangles, Descending









Appendix C

Wavelet Family	Resolution threshold	Threshold value	Accuracy	Total number of pattern discovered	Processing Time
Haar (DB1)	4	0.3	5.1%	9210	312s
		0.2	6.8%	7562	
		0.15	12.1%	2434	
		0.1	35%	652	
	5	0.3	6.1%	8234	931s
		0.2	7.2%	7632	
		0.15	14.3%	2215	
		0.1	42.5%	433	
	6	0.3	7.3%	7541	3143s
		0.2	9.5%	5432	
		0.15	21.3%	1953	
		0.1	51.2%	214	
	7	0.3	8.8%	6249	8328s
		0.2	9.4%	4528	
		0.15	23.7%	1659	
		0.1	40.5%	192	
Daubechies (DB2)	4	0.3	6.2%	8932	312s
		0.2	7.1%	7419	
		0.15	14.2%	3936	
		0.1	43.1%	543	
	5	0.3	7.1%	7734	931s
		0.2	9.4%	6498	
		0.15	17.4%	2096	
		0.1	53%	420	
	6	0.3	8.9%	7146	3143s
		0.2	13.5%	5942	
		0.15	19.9%	1873	
		0.1	56.9%	231	
	7	0.3	10.5%	6023	8328s

		0.2	14.5%	5129	
		0.15	18.5%	1543	
		0.1	48.3%	194	
Coiflet (C1)	4	0.3	6.6%	9212	312s
		0.2	7.2%	7511	
		0.15	15.1%	4217	
		0.1	44.3%	659	
	5	0.3	7.1%	7437	931s
		0.2	9.5%	6108	
		0.15	16.7%	1942	
		0.1	52.3%	431	
	6	0.3	8.6%	7106	3143s
		0.2	13.2%	5918	
		0.15	19.9%	1764	
		0.1	53.5%	253	
	7	0.3	9.6%	6023	8328s
		0.2	12.5%	5129	
		0.15	17.1%	1211	
		0.1	46.6%	152	
Symmlet (S8)	4	0.3	3.2%	8902	312s
		0.2	5.2%	6578	
		0.15	16.9%	3017	
		0.1	33.6%	798	
	5	0.3	8.8%	8176	931s
		0.2	9.5%	7016	
		0.15	15.1%	3512	
		0.1	45.8%	632	
	6	0.3	7.6%	6790	3143s
		0.2	12.6%	4378	
		0.15	20.7%	1549	
		0.1	52.6%	213	
	7	0.3	12.5%	5763	8328s
		0.2	16.1%	4092	
		0.15	19.9%	1214	
		0.1	53.2%	144	