CHAPITRE 5

FORECASTING WITH CI TECHNIQUES

IV estimation enables one to forecast volatility directly at the horizons of interest, without making assumptions about the nature of the volatility process. Christoffersen (2002) has further shown that the short-term volatility could be relatively accurately forecast *e.g.,* for a time horizon of five to ten working days. On the other hand, long term volatility forecastability seems to subject to some debate. Christoffersen and Diebold (1998) found that volatility forecastibilty declines quickly with horizon, and seems to have largely vanished beyond horizons of ten or fifteen trading days, whereas Ding, Granger and Engle (1993), Baillie *et al.* (1996) and Andersen *et al.* (2001, 2001b) claimed that volatility displays long memory.

The possibility of market timing focuses, not on prediction of returns directly, but rather on prediction of *signs* of returns, on the grounds that profitable trading strategies may result if one is successful at forecasting return signs, quite apart from whether one is successful at forecasting the mean of returns. A well-known and classic example involves foreign exchange trading. If the Yen/$ exchange rate is expected to increase, reflecting expected depreciation of the Yen relative to the dollar and hence a negative expected "return" on the Yen, one would sell Yen for Dollars, whether in the spot or derivatives markets (Christoffersen and Diebold, 2002). From the viewpoint of trading VIX and the associated futures, such an approach is also applicable for the volatility forecasting. In this chapter and in the following one, we attempt to forecast both the direction and range of the volatility by accounting for the non-linearity of the IV in a relatively short time span, *e.g.* within five working days. More specifically in the last part of this chapter, the actual value of the volatility will be forecasted based on a similar approach.

5.1 IV Calculation

The first step is to calculate the IV of the S&P100 index between Jan. 2, 1998 and Aug. 29, 2003, a series containing 37,995 entries. The return of an asset r at time t_i is defined as

$$r(t_i) = r(\Delta t; t_i) = x(t_i) - x(t_i - \Delta t) \tag{16}$$

where $x(t_i)$ is a homogeneous sequence of logarithmic prices as defined by

$$x(t_i) = x(\Delta t, t_i) = \tfrac{1}{2}\left(\log p_{\text{bid}}(t_i) + \log p_{\text{ask}}(t_i)\right) \tag{17}$$

where $p_{\text{bid}}(t_i)$ and $p_{\text{ask}}(t_i)$ are the bid and ask prices of the underlying asset at time t_i.The IV is evaluated based on intra-day historical data and is a more accurate approximation of the daily volatility (Andersen and Bollerslev 1998) :

$$\nu(t_m) = \nu(\Delta t, n; i) = \left[\frac{1}{n}\sum_{j=1}^{n}\left|r(\Delta t; t_{i-n+j})\right|^2\right]^{1/2}, \tag{18}$$

where $\nu(t_m)$ is the moment rate of return distribution, Δt is the time interval of the data in which integration is done, n is the total time length of the integration, i is the total number of data and p defined the $(1/p)^{th}$ moment.

In the GA case detailed in Section 5.4, the S&P100 index 15-minute interpolated time series, Δt = 15, n = 28, $1/p$ = 2, and $m = i/n$ (where m =1, 2, ..., 1350). In other words, one deals with m days of data with 6.5-hour usual trading time per day. The hence derived IV series could be de-trended through different ways of normalizations *e.g.*, take logarithm of the data and subtract all by the mean (Kinlay, 2001). In the current study, the main interest is to determine the short term abrupt changes in the IV series, which is the key difficulty confronted the contemporary researchers. Volatility term structures are normally mean-revert, with short-term volatility lying either above or below the long term mean, depending on whether current conditions are high or low volatility. Therefore, the IV series is normalized by its past 21 day moving average, which roughly

encompassing the immediate past month trading. The one-month time horizon is for the convenience of trading index options.

5.2 Wavelet Transform

Prior to analysis, the data representation and/or pre-processing are often necessary before actual data mining operations can take place. The representation problem is especially important when dealing with time series, since direct manipulation of continuous, high frequency data in an efficient way is extremely difficult, although the current case is a one-dimension series. This problem can be addressed in different ways. One possible solution is to use windowing and piecewise linear approximations to obtain manageable sub-sequences. The main idea of transformation based representations is to transform the initial sequences from time to another domain, and then to use a point in this new domain to represent each original sequence. Wavelet analysis is a form of non-parametric regression analysis. It extracts both low and high frequency components of a given signal. It has found much success in applications such as image processing, geological testing and many other engineering applications.

Specifically, the Discrete Wavelet Transform (DWT) translates each sequence from the time domain into the time/frequency domain. The DWT is a linear transformation, which decomposes the original sequence into different frequency components, without loosing the information about the instant of the elements occurrence. A wavelet transform is a scaling function used to convert a signal into father and mother wavelets. Father wavelets are representations of a signal's smooth or low-frequency components. Mother wavelets represent the details or high-frequency components in the signal (Kaboudan, 2005). The sequence is then represented by its features, expressed as the wavelet coefficients. Again, only a few coefficients are needed to approximately represent the sequence. With these kinds of representations, time series becomes a more

manageable object, which permits the definition of efficient similarity measures and an easier application to common data mining operations (Antunes and Oliveira, 2002).

As indicated in Chapter 2 however, the wavelet methodology has not been used widely in economic time series studies. So far only a few researchers make use of wavelets to analyze individual stock returns, high frequency stock index returns (Capobianco, 1999, and Arino, 2000) and the foreign exchange rates (Los and Karuppiah, 2000, Kaboudan, 2005). Hog and Lunde (2003) made used of wavelet transform to estimate IV of the foreign exchanges, but did not touch on the topic of forecasting.

In this research work, the proposed methodology uses wavelets to decompose the series into superposed spectral components. The nonlinear time-varying underlying trends in noisy series, such as the clustering phenomenon of the current volatility series, can be identified. After the process, the coefficient of each scale can provide clues for the pattern of the volatility in the corresponding time horizon. For example, long-term trends can be assessed from the coarse-scale behaviour of the series. Shorter-term trends and noise can be estimated from fine-scale behaviour. In a decomposed wavelet plot, if the volatility at a certain scale is high, we might hypothesize that high volatility is also likely to appear in the next few components at and/or near the same scale, because of the clustering effect. It will be interesting to confirm such a hypothesis by finding out that the high values indeed appear at the *same scale* in the subsequent time intervals. If so, when a high volatility appears, one could reasonably expect that high volatility would also occur in a time horizon that is associated with the found scale. Therefore, a qualitative forecasting on a component basis will be of practical value in the process of building a quantitative forecasting algorithm.

Once the time series is converted to the normalized IV's, it will be analyzed with the DWT, which decomposes the original sequence V into different frequency components without loosing the information about the instant of the elements occurrence. The

sequence is then represented by its features, expressed as the wavelet coefficients $b_{j,k}$ and $c_{j,k}$. Refer to APPENDIX 3 for details. Since only a few coefficients are needed to approximately represent the sequence, time series becomes a more manageable object, which permits the definition of efficient similarity measures and an easier application to common data mining operations (Antunes and Oliveira, 2002).

The main advantage when using the wavelet method is its robustness due to the absence of any potentially erroneous assumption or parametric testing procedure. Another advantage is that wavelet variance decomposition allows one to study different investing behaviour in different time scales (horizons) independently. Different investing styles may cluster into different time horizons. In wavelet packet trees, those IV values that occur often *i.e.,* at higher probability correspond to nodes at a higher level in the tree (Mallat, 1999). In the best tree, only those IV patterns that are uniquely decodable get to the bottom of the tree. Other repetitive patterns get located closer to the root. The root node represents the original data sequence, which would mean daily forecast and thus provide no calculation economy.

Another property attribute is the de-correlation, or so called whitening, which means correlated signals in time domain become almost uncorrelated coefficients in the time-scale domain. And due to the inherent characteristic property that wavelets are very good at compressing a wide range of signals into a small number of large wavelet coefficients, a very large proportion of the coefficients can be set to zero without any loss of important embedded information. This property allows wavelets to deal quite well with heterogeneous and intermittent behaviors (Silverman 2000). Therefore, wavelet transform can be utilized to filter out "noise" traders (wavelets compressing) and separation of short term and long run performances (time-scale decomposition). In summary, wavelet method has the following property attributes for our purpose:

a) Perfect reconstruction if and when needed

b) Locality in time and in scale
c) Whitening
d) Dis-balance energy
e) Filtering
f) Detect self-similarity
g) Efficient algorithm capability (Li, 2003)

5.3 Wavelet Packets

Details regarding the concepts and advantages of applying wavelet packets, thresholding and filtering to analyze nonlinear time series are provided in the following statements. Wavelet packet analysis provides richer details compared with simple DWT. In wavelet packet trees, those IV patterns that occur often, *i.e.* at higher probability correspond to nodes at a higher level in the tree (Mallat, 1998).

The wavelet packets can help differentiate the signal in a wider range of scales and it could identify the coefficients where patterns repeat the most so that the analysis could be more focused. By employing the *besttree* function, one could pinpoint the best way to dissect the original series and find out the best forecasting frequency. For example, the *besttree* function returns a tree with high repetition rate at node [2, 0], so that one only needs to focus on the eight-day forecasting range instead of daily, an eight-time economics in computation time. Moreover, a better forecasting accuracy could be expected due to the focus at the range where higher concentration of energy locates. The general criteria in selecting the forecasting range are listed below as a sequence of steps:

Run the wavelet packet analysis and find the best tree by selecting the number of levels based on analyst's interest and by selecting the order of the filters in order to eliminate maximum amount of data without affecting the reconstruction while matching most patterns of the original data;

Find the entropy values associated with all terminal nodes on the best tree;
Concentrate on the terminal nodes with the low entropy values, knowing that the original node carries the highest value.

In this research thesis, the number of level of the wavelet packet as well as the order of filter is determined through a trial and error process. Once the best tree is selected based on the entropy value, the corresponding wavelet coefficients could be analyzed with GA programs and are described in the next section.

5.3.1 Strategic Deployment of some Wavelet Packet Techniques

The simplest wavelet non-linear compression technique is thresholding (Donoho 1995). Some recent work can be found in (Donoho and Johnstone 1998, Donoho and Yu 1998, and Donoho and Johnstone 1999), where statistical optimality of wavelet compression was explored, and Bayesian approaches are incorporated in selections of the compressing thresholding rules. The commonly used criterion for choosing the most efficient or best basis (pattern) for a given signal is the minimum entropy criterion (Coifman and Wickerhauser, 1992 and Wickerhauser, 1994). That is, let $\{p_i\}$ be the decomposition coefficients of a signal for a particular choice of the wavelet packet basis. For each set of decomposition coefficients $\{p_i\}$ we associate a nonnegative quantity $\eta^2(\{p_i\})$ called entropy defined by

$$\eta^2(\{p_i\}) = -\sum_i \frac{p_i^2}{\| p \|^2} \log_2 \frac{p_i^2}{\| p \|^2} \tag{19}$$

where $\|p\|^2 = \sum_i p_i^2$. The best basis is the one which produces the least entropy. Intuitively, the entropy defined above gives a measure of how many effective components are needed to represent the signal on a specific basis. For example, if in a particular basis the decomposition produces all zero coefficients except one *i.e.,* the signal coincides with a wave form, then the entropy reaches its minimum value of zero. On the other hand, if in some basis the decomposition coefficients are all equally

important, say $p_i = 1/\sqrt{N}$ where N is the length of the data, the entropy in this case is maximum, $\log_2 N$. Any other decomposition will fall in between these two extreme cases. In general, the smaller the entropy the fewer significant coefficients would be needed to represent the signal.

In using the one-dimensional wavelet packet compression approach, the current research effort employs the global soft thresholding method (Mallat, 1998). Predefined thresholding strategy is based mostly on empirical methods – to strike a balance between sparseness and details. For example, in retaining 90% of the entropy, a 70-80% of data would have been removed which in turn results in the data compression.

The type of filter to be deployed could be selected based on the nature of the data set *e.g.,* for time series with sharp jumps or steps, one would choose a boxcar-like function such as Harr (Torrence and Compo, 1998). Daubechies wavelets filters are optimal in the sense that they have a minimum size support for a given number of vanishing moments. When choosing a particular wavelet, one thus faces a trade-off between the number of vanishing moments and the support size. If V has few isolated singularities and is very regular between singularities, one must choose a wavelet with many vanishing moments to produce a large number of small wavelet coefficients $<\theta, \theta_{j,n}>$. If the density of singularity increases, it might be better to decrease the size of its support at the cost of reducing the number of vanishing moments. Indeed, wavelets that overlap the singularities create high amplitude coefficients (Mallat, 1998).

5.4 Forecasting by Genetic Algorithms

Review of technical literature on applying a variety of techniques including the GA/GP to forecast volatility reveals that except those based on proprietary methodology which is not available in the public domain, researchers still rarely have found any effective and systematic method to deal with non-linearity. Moreover, very little research has

been attempted to take advantage of the recent advances in the estimation of volatility, namely to couple the calculation of IV with those dealing on non-linear analytical techniques such as GA/GP and wavelet transforms. By applying the GA approach, the nonlinear time-varying underlying trends in noisy series, such as the clustering phenomenon as well as sudden jumps and drops in the volatility series can be properly identified. Since no pre-assumption is needed to be made and since both wavelet transformation and GA are non parametric, the current IV-wavelet-GA method is therefore inherently more robust, accurate and efficient to yield an effective solution for use in forecasting financial index volatility compared with any of the prevalent methods of today. In the following sections, the proposed IV-wavelet-GA approach is explained and is followed by a description of the investigation to forecast the short term moving direction and range of IV of the S&P100 index.

As indicated earlier, GA's can optimize some arbitrary function with straightforward representation better than many other procedures. Simplicity of operation and power of effect are two of the main attractions of the GA approach. GA's efficiently build new solutions from the best solutions of the previous trials, regardless of the linearity conditions of the problem on hand (Bauer, 1994). Moreover, GA's may help uncover those hidden rules, either general or specific for the asset besides the well documented clustering effect for the volatility such as crucial non-linear regularities.

5.4.1 Problem Formulation

As indicated in Chapter 1 and APPENDIX 2, a classical linear expression or factor format expression of GARCH explains mainly the clustering effect, but inherently lacks in accuracy when dealing with non-linearity. Andersen *et al.* (1998) introduced the concept of IV and improved the forecasting confidence of the one-day-ahead volatility. They made use of the diffusion limit of the weak GARCH(1,1) process to construct the continuous time model of daily volatility σ_t:

$$d\sigma_t^2 = \xi\left(\omega - \sigma_t^2\right)dt + \left(2\lambda\xi\right)^{1/2}\sigma_t dW_{\sigma,t} \tag{20}$$

where $W_{\sigma,t}$ is the generalized Wiener process and ω, λ and ζ can be expressed in terms of the discrete-time weak GARCH(1,1) parameters. Such a weak GARCH(1,1) model converges to an IGARCH(1,1) as the sampling frequency increases. Andersen (2001) ruled out any sizable improvement of predictability by means of higher order discrete time ARCH approximations or by using more complicated stochastic differential equations, due to the weak GARCH(1,1) interpretation of diffusion approximation in Eq. (20). However, the derivation of IV converts the volatility from a latent variable into an "observable" one. As a result, any volatility forecasting could be performed based on more realistic historical data.

5.4.2 Proposed Methodology

The research attempted in this part of the thesis work may be represented in terms of the process modules as shown in Fig. 3.

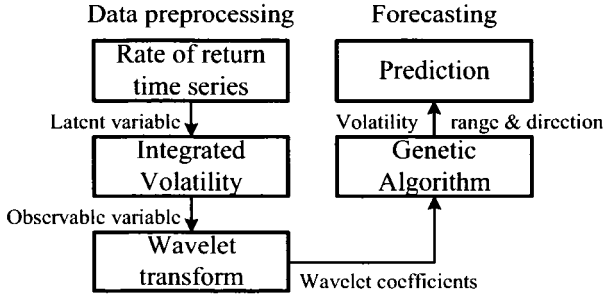

Figure 3 Proposed volatility forecasting methodology

Details about the implementation of this proposed methodology are described in the following sections.

5.4.3 Parameters Selection

One advantage of combining wavelet analysis with GA is the flexibility that they bring in. When applying the GA technique in the following sections, as mentioned before, we will adopt a 4-lag recursive forecasting approach. By selecting the corresponding wavelet coefficient series, this 4-lag configuration could help focus on different time ranges depending on nodes on the wavelet tree as given by the following expression:

$$\frac{f_{j,n}}{\Delta t} = \left[\frac{n}{2^{j+1}\Delta t}; \frac{n+1}{2^{j+1}\Delta t} \right], \tag{21}$$

where j is the level on the tree, n is the location on the tree, Δt is the sampling period; in the current case, it is one day and $f_{j,n}$ is the nominal frequency band.

For example, wavelet packet node (1, 0) gives information up to four days ahead; while (5, 5) is between 11 and 13 days. As a result, one could concentrate on uncovering trading patterns of different horizons.

For a data set of N samples of IV's, level 2 of the wavelet packets has $N/2$ number of coefficients, representing a saving of 50% calculation for the subsequent GA processing. A maximum of level 5 has been selected as the analysis scale here, because as mentioned earlier reliability of forecasting accuracy drops as time horizon expands. Filters such as the db2 wavelet, which has two vanishing moments, were used for the current analysis in order to maximize the match of the reconstructed data with the original data while retain the minimal amount of data. Note that a wavelet of the Daubechies family with fewer vanishing moments may fail to suppress the higher order polynomial signal. This has been confirmed in analyzing the current S&P100 series when db1, db2 and db3 occasionally fail to generate the wavelet coefficients based on the best tree that is created from the wavelet packet. On the other hand, higher order wavelet tends to generate smoother decomposed plots, which may loose some desirable details from the original series. Different combinations of orders and levels of the db

wavelets could be tried to obtain the best tree. Analysis could focus on the node with the lowest entropy. For example, db3 with level 5 in the best tree, the lowest entropy occurs in packet (4, 0), where $j = 4$ and $n = 0$. Packet (4, 0) corresponds to the frequency of $(n + 1)/(2^{j+1} \Delta t) = 1/(32) \Rightarrow 32$ days (in case of $j = 1$, $n = 2$, $(2 + 1)/(4 \times 1) \geq 1.3$ days). In general, there are five parameters to be determined before conducting the analysis *i.e.*, number of levels of the wavelet packet tree, order of the filter, number of generations, number of groups of rules and the training period. In this thesis research study, the effect of each variable is investigated in a sequence of comparative analysis by holding others temporarily constant. Since the main difficulty that many contemporary researchers were facing is the forecast of abrupt changes, the short term patterns in the volatility are the primary focus of this research investigation.

5.4.4 Implementation of the GA Forecasting Process

The premise of the GA approach described in this chapter is based on the concepts derived from Fong and Szeto's method (2001). First, we de-trend the wavelet coefficients with a 21-day-moving-average operation, convert them into integers of one to four corresponding to the four selected ranges, and then generate the rules randomly in the form as was shown in Section 3.3.1 earlier. For the '`THEN`' part of the rules, there are four different classes, 1, 2, 3 and 4. We randomly generate 25 rules for each class to have a total 100 rules. Then we repeat the process to generate 100 groups of such rules.

Fitness value of each rule is calculated as described in the following paragraphs. In each training step, the rules for class k are trained by comparing the patterns of the randomly generated rules with the S&P100 IV historical data. Three possible cases can arise. They are:

Case 1: the '`IF`' part of the rule does not match the data point pattern. So, no prediction can be made.

Case 2: the 'IF' part of the rule matches the data points in the training set. Prediction can be made. When the 'THEN' part of the rule also matches the class of the corresponding data point, it is counted as a correct guess otherwise a wrong guess. The fitness value of rule *i*, *i.e.* **the forecasting accuracy** will be

$$F_i = N_c / N_g = N_c / (N_c + N_w). \tag{22}$$

Here N_c is the number of correct guess and N_w is the number of wrong guess, so that

$$N_g = N_c + N_w. \tag{23}$$

We apply each rule to all training data and find the accumulated N_c.

Case 3: there are more than one rule with the 'IF' part, which matches the data points in the training set. The most specific rule, which does not have "don't care" and all logical operators are 'AND', is chosen.

These rules will be ranked based on their fitness for the subsequent self-reproduction process. We then repeat these steps sequentially throughout the training data set for other 99 groups. Out of the 50 groups with F_i's below its medium, we randomly choose 25 groups for crossover, in which each group goes through the following process :

a) From the pool of 100 rules, we randomly select 2 rules to conduct crossover;
b) We register the rules in a memory;
c) From the second round of selection onward, we compare selected the rules with those stored in the memory;
d) If both rules have been selected as a pair before, then we repeat the selection process till picking a different pair. We repeat the process until we have formed 100 new rules.

The other 25 groups undergo mutation with at a rate of 4%, which means 1% overall in each generation. The same process is repeated for a preset number of generations, *e.g.*

1000. In each generation, only 50% of rules need to be evaluated for F_i's, therefore resulting in a 50% of CPU time saving. At the end, the best group of rules is selected for the purpose of testing of their forecasting accuracy with the subsequent part of the data.

5.4.5 Testing of the Methodology and Results

The intraday data of S&P100 between 1987 and Aug. 2003 was acquired from TickData Inc. Part of the data set, the 15-minute high-low prices between 1998 and 2002 is taken for the training purpose. The second part *e.g.*, between Jan. 2 and Aug. 29, 2003 is used to test the validity of the rules. The data are imported into a Matlab environment to calculate the corresponding normalized IV's. The IV values are then wavelet transformed to find the best tree. The GA programs are then applied to forecast the IV values at the selected time ranges ahead. In applying the GA programs, all rules are initially assigned to have zero fitness. The data range. $(-\infty,-a]$, $(-a, b]$, $(b, c]$, (c, ∞) are preset at $a = -0.3$, $b = 0$ and $c = 0.3$ to evenly distribute data into four ranges and to meet analyst's risk requirement. The data is then processed with the GA programs and the best group of rules is found. Upon completion of the training process, the best group of rules will be used to test the subsequent part of the S&P100 IV data to assess the forecasting accuracy, *i.e.* if each of the forecasted one-step-ahead point is at the same range as the actual data. To achieve calculation economy, the programs that involve GA are written in Java while the wavelet transformation is performed with Matlab. It could be observed from Fig. 4 and 5, (while it may be less obvious in Fig. 6 that

a) the forecasting accuracy is generally above 60% shown at the Y-axis in the respective figures, which is better than the traditional methods and matches those derived from the proprietary methods (Kinlay, 2001);
b) the forecasting accuracy is higher for the wavelet transformed series with higher scales compared with those derived on the original series *i.e.,* the non-transformed

ones. This may be attributed to the fact that variance of the original series is the sum of variances of its spectral components.

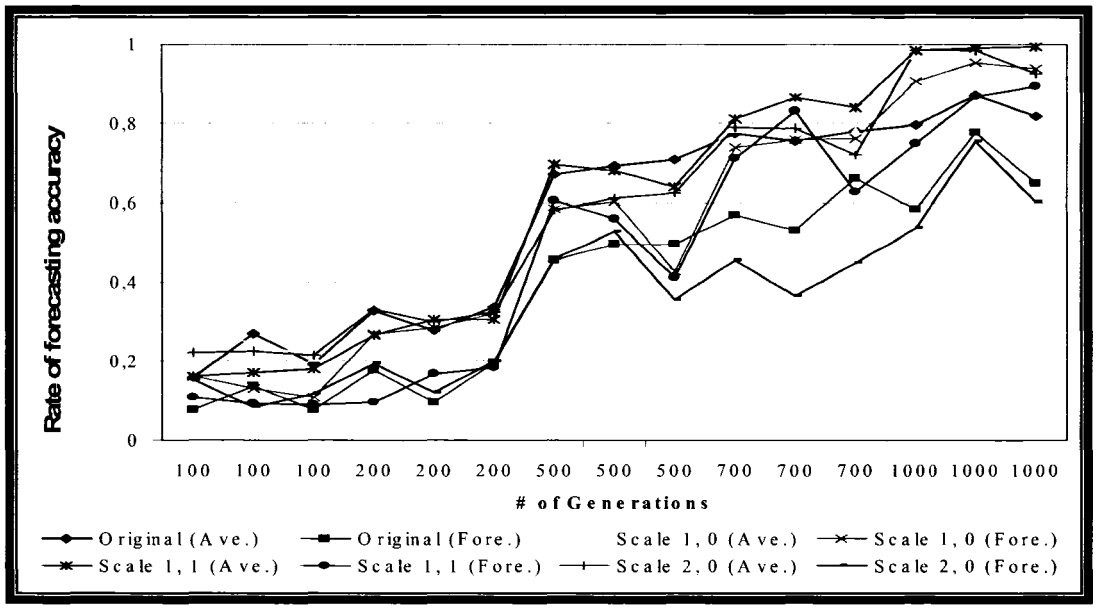

Figure 4 Daily 2003 S&P100 forecasting accuracy based on 2002 S&P100 data

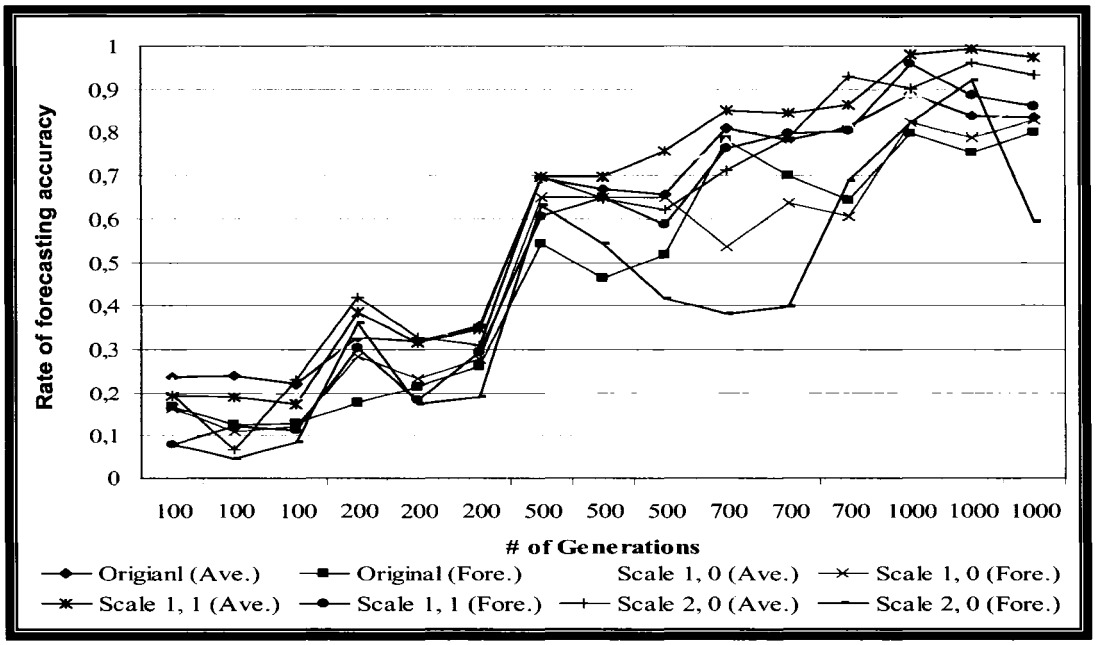

Figure 5 Daily 2003 S&P100 forecasting accuracy based on 2001-2 S&P100 data

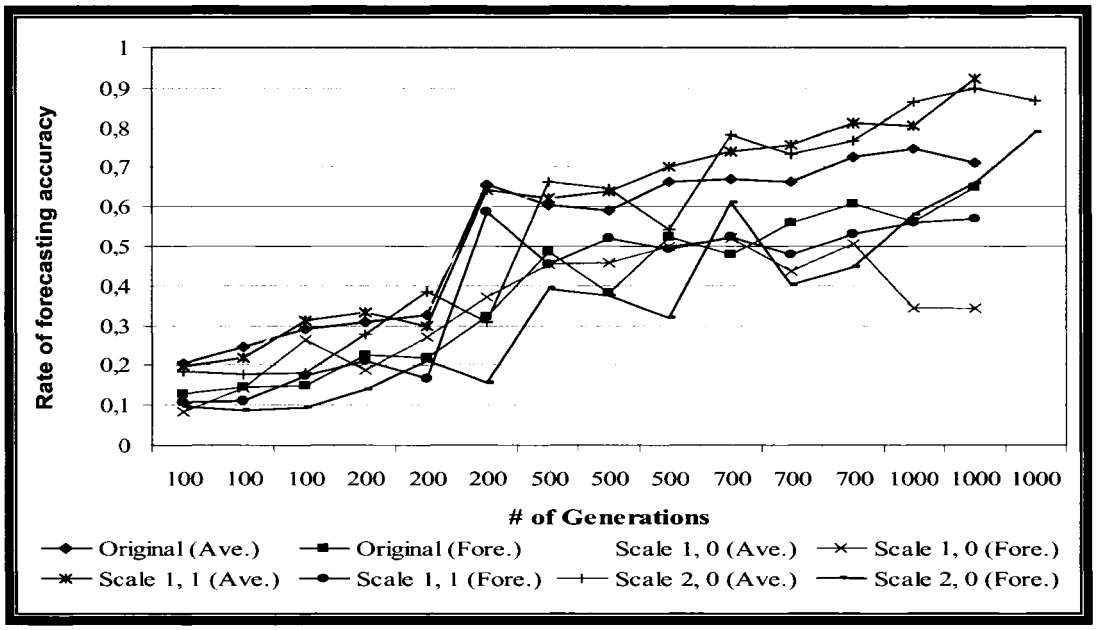

Figure 6 Hourly 2003 S&P100 forecasting accuracy based on 2002 S&P100 data

The same data in the same one-year, two-year and five-year time horizons as shown above in the current research are also processed with the GARCH(1,1) model. The forecasting values from GARCH(1,1) are first normalized to the respective logarithmic means and are then converted into values of 1, 2, 3 and 4 according to their amplitudes, similar to the preprocessing described in the previous section where GA's are used. The accuracy of forecasting is calculated based on the comparison between the converted data and the realized volatility in the validity-testing period *i.e.*, from Jan. 03 to Aug. 29, 2003. The one day ahead forecasting accuracy for the 2003 S&P100 daily data based on training sets at the selected periods are summarized as shown in Table II.

Table II

The one-day-ahead GARCH forecasting accuracy

Period	% Accuracy
2002	0.485
2001 – 2002	0.491
1998 – 2002	0.503

They agree well with the results derived in many contemporary GARCH as well as IV studies (Andersen, 1998, 2001), but are markedly lower than those achieved by using the GA method proposed and developed in this research thesis. It may be argued the GA method is superior to the GARCH approach simply because it takes more historical patterns linear or nonlinear, into consideration for forecasting purposes. Our current method takes the past four data points joined with the logical arguments '`AND`' to form 1024 possible combinations in a training data set to find a relatively optimal group of rules. By taking the '`OR`' argument and "`don't care`" class, even more combinations have been included.

In many contemporary research publications dealing on volatility estimation and forecasting, the dominant evaluation criterion is the coefficient of determination R^2 (multiple correlation coefficient), which simply represents the fraction of variability in y (the linear regression function) that can be explained by the variability in x (regressor) (Taylor, 1999). In other words, it explains how much of the variability in the y's can be explained by the fact that they are related to x *i.e.*, how close the points are to the line. Since we are not using a linear model in the current approach and since Taylor (1999) has demonstrated some of the drawbacks of using R^2, we adopt instead the percentage of correct forecast as the measure of accuracy.

5.4.6 Discussions on the Present Methodology

The use of the GA four-lag forecasting algorithm is based on the stylized fact of volatility clustering that has been the foundation of the GARCH approach. This method helps analysts identify different patterns in the time domain even if those patterns are abrupt jumps or drops. Since it is non parametric and free of any pre-assumption, it is more flexible and robust to deal with non-linearity. The wavelet transform enables analysts to study the volatility patterns in different frequencies (time horizons). The combination of these two techniques opens up a broader field for analysts to explore different properties hidden in the volatility series. The fact that for some time horizons the forecasting accuracy is higher for the wavelet transformed series with longer time ranges implies that it could be beneficial for analysts to focus on those time ranges with lower entropy.

With such a speedy processing, one could afford to increase the number of lags from four to five or even higher, and increase number of the amplitude ranges from the current four to six to be more precise in forecasting the volatility values. In such a case, the need to de-trend the data by normalizing the data with the 21 day moving average could be diminished.

5.5 Forecasting by Genetic Programming

In this Section, we intend to take the forecast process one step further – instead of using GA's to forecast the moving direction and range of IV, we will apply GP to forecast the value of IV. GP was first developed as an extension to GA's. The most important feature of GP is their representation of individual solution structure. Unlike traditional GA's, which usually represent individuals as vectors of fixed length, GP individual is represented as hierarchical composition of tree-like structure with variable length from basic building blocks called functions and terminals. The function set is composed of the

statements, operators, and functions available to the GP system. The terminal set is comprised of the inputs and constants to the GP program. GP possesses no inherent limitations on the types of functions, as long as the closure property is satisfied, that is each function should be able to handle gracefully all values it might receive as inputs.

Traditionally, GP uses a generational EA. In this approach, there exist well-defined and distinct generations. Each generation holds a complete population of individuals. The newer population is created from the genetic operation on the older population and then replaces it (Chen, 2003). GP-resulting specifications may be viewed as coincidental equations that may capture the dynamics of a process. Coefficients in GP-evolved equations are not computed but randomly generated. It is therefore, advantageous over the conventional statistical regression, *e.g.* robust against problems of multicolinearity, autocorrelation and heteroscedasticity. There are also no degrees of freedom lost to compute the coefficients. However, because variables and operators selected to assemble the equations are random, while attempting to breed the fittest individual equation, the program occasionally gets trapped in local optima rather than a global one within the search space. It is therefore, necessary to generate a large number of equations and then select the best ones for forecasting as suggested by (Kaboudan, 2005).

GP is fundamentally a computer search and problem-solving methodology that can be easily adapted for use in non-parametric estimation. It has been shown to detect patterns in the conditional mean of foreign exchange and equity returns that are not accounted for by standard statistical models as shown in the corresponding references listed in Table 2.1 and others (Neely, Weller, and Dittmar 1997; Neely and Weller 1999; Neely 2000). This suggests that a GP based method as presented here may also serve to be a powerful tool for generating predictions of asset price volatility. For a summary of the basic concepts of GP, please refer to APPENDIX 6 at the end of this thesis.

5.5.1 Forecast of IV of a Financial Time Series

As indicated by Radzikowski (2000), modern parametric option-pricing models, where volatility is often the only stochastic variable, were expected by many researchers as well as end-users to :

a) Be well-specified,
b) Consistently outperform other models,
c) Be statistically consistent with underlying asset return dynamics,
d) Provide a statistical theory of option pricing error, and
e) Be elegant and not difficult to estimate

But they have uniformly failed to deliver against these expectations, as they either are too complex, have poor out-of-sample performance, make unrealistic distribution assumptions, and/or use implausible and/or inconsistent implied parameters. While parametric models provide internal consistency, they do not out-perform simplistic approaches out-of-sample. Even the most complex modern parametric models are imperfect and are outperformed by simple, less general models. Jackwerth and Rubinstein (2001) applied series of tests to a variety of models and concluded that naïve approaches are consistently the best, stochastic deviation models are the next best, then there are deterministic volatility models that follow and then comes finally the traditional parametric models.

5.5.2 Problem Statement of IV Forecast

In this Section, we will attempt to forecast the numerical values of the volatility by formulating a nonlinear and non parametric approach based on GP in the TSDM framework. Different patterns, linear or non-linear including the stylized clustering effect of volatility may repeat in different time intervals. This is true when dealing with

different types of financial securities or dealing with different historical periods for the same underlying security. By making use of the stylized characteristics of financial volatility, we extend the TSDM method with GP to forecast as many events/non-events as practically feasible in the IV time series in order to guide option trading.

5.5.3 Data Analysis and Final Results

The intraday data of S&P100 index between 1987 and Aug. 2003 is acquired from TickData Inc., and the 15-minute high-low prices between Dec. 3, 2001 and Dec. 31, 2002 are taken for our training purpose. The second part, *e.g.* between Jan. 2 and Aug. 29, 2003 is utilized to test the validity of the rules. The first 21 days of both sets of data are used to prepare for the 21-day moving average, in order to take the monthly effect into consideration, to de-trend (or normalize) and to improve the forecasting accuracy. The corresponding normalized IV's were then calculated and fed into the GA's to forecast the moving directions and to find the best 100 rules by maximizing the value f (Ma *et al.*, 2004b). The GP programs are then applied to forecast the IV values at the selected time ranges ahead, *e.g.,* one period ahead.

The execution cycle of the generational GP algorithm consists of the following steps:

1). First, we initialize the population. An initial population of 100 is created randomly from the basic building blocks.
2). We then evaluate the individual programs in the existing population. A value for fitness, *e.g.* the absolute difference between the individual and the desired one is assigned to each solution depending on how close it actually is to solving the problem (thus arriving to the answer of the desired problem).
3). Until the new population is fully populated, we repeat the following steps:
    a. We select an individual or individuals in the population using the selection algorithm

b. We perform genetic operations (crossover & mutation) on the selected individuals

c. We insert the result of the genetic operations into the new population.

4). If the termination criterion is fulfilled, then we continue. Otherwise, we replace the existing population with the new population and repeat steps 2-4

5). We can then present the best individuals in the population as the output from the algorithm.

Table III

GP Configuration

Parameter	Values
Generations:	25/50/100
Populations:	100
Function set:	+, –, %, *, sin, exp, sqrt, ln
Terminal set:	$\{x(t\text{-}1), \ldots x(t\text{-}4)\}$
Fitness:	difference between actual and desired
Max depth of new individual:	6
Max depth of new subtrees for mutation:	6
Max depth of individuals after crossover:	17
Muataion rate:	0.05
Generation method (selection):	50%

Table III lists the parameters used in this research. Note that based on the findings in Neely and Weller's research publication (2001), the fitness of the GP operation in the current investigation is derived from the Mean Absolute Error (MAE) between the generated individual and the actual IV value. As found by Park (2002) while testing an MAE based GARCH model (so called robust GARCH), MAE tends to generate results superior to GARCH models that are MSE based. Two separate tests have been conducted on the 2002 training data set by pre-processing them with the GA's (Ma *et al.*, 2004*b*), one for 500 generations and the other 1000. The intermediate results are

then passed through GP programs and the final results of percentage accuracy are randomly selected and shown in Table IV. For example, an initial population of 100 rules is generated and 25 generations of GP are performed with a maximum depth of six of new individuals.

Table IV

Random samples of forecasting accuracy for 2003 IV data (2002 training data set)

GP Parameters	2002 data set (500 generations GA)	2002 data set (1000 generations GA)
[25, 100, 6]	72.65, 73.21, 74.40	74.23, 66.00, 68.77
[50, 100, 6]	71.46, 75.36, 74.46	76.77, 69.13, 67.54
[100, 100, 6]	71.44, 69.60, 68.42	68.09, 67.49, 67.17

The 2001-2002 training data set was then pre-processed using GA's (Ma *et al.*, 2004*b*) and 1000 generation GP was implemented to obtain the results as shown in Table V. These results are randomly selected from the respective test groups for sake of brevity.

Table V

Random samples of forecasting accuracy for 2003 IV data (2001 – 2002 training data sets)

GP Parameters	2001/2002 data (based on 1000 generations GA)
[25, 100, 6]	78.25, 77.66, 78.25
[50, 100, 6]	80.20, 79.51, 79.54
[100, 100, 6]	79.10, 78.86, 78.98

An interesting phenomenon could be observed that the forecasting accuracy in the current tests is not positively related to the number of generations used in either GA's or the subsequent GP operations. This may be caused by the early convergence to the local minima in the search process. Further investigation and appropriate search strategy may be necessary to resolve the issue.

5.5.4 Discussions

By working in the novel TSDM framework with the associated methods, tests conducted on the proposed methodology developed in this research have made use of GP to find optimal temporal pattern clusters that both characterize and predict time series events. Additionally, a time series windowing techniques is adapted to allow prediction of non-stationary events. Results of the tests have demonstrated that the modified TSDM framework successfully characterizes and can be relied on to predict complex, non-periodic and irregular IV time series. This was done through testing the S&P100 index of different years. The one step ahead forecasting accuracy reaches an average of 74% with standard deviation of 4.6%. This accuracy as well as reliability of forecast in market applications is considered well above average.

The forecasting accuracy achieved with GP is, however, somewhat lower than those derived by GA's not only in terms of absolute level but also on the trend of convergence. As shown in Table V, accuracy did not improve with the increase of the number of generations. One reason may be that the forecasting accuracy of GP in the current case is built upon results of the GA processing. Errors from the GA part may affect the subsequent GP operation. This simply demonstrates the high level of difficulty of pinpointing a value of future IV. And it leads us to believe that some more work needs to be carried out before this part of the program can be successfully used in practice. Local search algorithms such as conjugate gradient technique proposed by Zumbach *et al.* (2001) described in Chapter 2 could be helpful in tackling problems such as this one. On the other hand, the GA part seems to be more robust and reliable, therefore it is recommended for immediate application. Future work in this regard may include, *a*) incorporating the modified TSDM*e2* based GP directly with wavelet transform, which might lead to improved forecasting accuracy, *b*) using parallel processing techniques to accelerate GP process, and *c*) testing other financial indices over a wider time span, *etc.*

These are just a few suggestions that may fill the missing links still left in the complex problem of market index forecasting.