Chapter 2
Self-adaptive Interval Type-2 Fuzzy Set Induced Stock Index Prediction

Abstract This chapter introduces an alternative approach to time-series prediction for stock index data using Interval Type-2 Fuzzy Sets. The work differs from the existing research on time-series prediction by the following counts. First, partitions of the time-series, obtained by fragmenting its valuation space over disjoint equal sized intervals, are represented by Interval Type-2 Fuzzy Sets (or Type-1 fuzzy sets in absence of sufficient data points in the partitions). Second, an interval type-2 (or type-1) fuzzy reasoning is performed using prediction rules, extracted from the (main factor) time-series. Third, a type-2 (or type-1) centroidal defuzzification is undertaken to determine crisp measure of inferences obtained from the fired rules, and lastly a weighted averaging of the defuzzified outcomes of the fired rules is performed to predict the time-series at the next time point from its current value. Besides the above three main prediction steps, the other issues considered in this chapter include: (i) employing a new strategy to induce the main factor time-series prediction by its secondary factors (other reference time-series), and (ii) self-adaptation of membership functions to properly tune them to capture the sudden changes in the main-factor time-series. Performance analysis undertaken reveals that the proposed prediction algorithm outperforms existing algorithms with respect to root mean-square error by a large margin ($\geq 23\%$). A statistical analysis undertaken with paired t-test confirms that the proposed method is superior in performance at 95% confidence level to most of the existing techniques with root mean square error as the key metric.

Abbreviations

CSV	Composite secondary variation
CSVS	Composite secondary variation series
FOU	Footprint of uncertainty
IT2	Interval type-2
IT2FS	Interval type-2 fuzzy set
IT2MF	Interval type-2 membership function
MFCP	Main factor close price
MFTS	Main factor time-series
MFVS	Main factor variation series

A. Konar and D. Bhattacharya, *Time-Series Prediction and Applications*,
Intelligent Systems Reference Library 127,
DOI 10.1007/978-3-319-54597-4_2

RMSE	Root mean square error
SFTS	Secondary factor time-series
SFVS	Secondary factor variation series
SFVTS	Secondary factor variation time-series
T1	Type-1
T1FS	Type-1 fuzzy set
VTS	Variation time-series

Symbols

$A_{i,j}$	Type-1 fuzzy set for partition P_i of MFTS
$\tilde{A}_i$	Interval type-2 fuzzy set for partition P_i in MFTS
B_i	Classical set for partition Q_i of MFVS
B'_i	Classical set for partition Q_i for SFVS/CSVS
$c(t)$	Close Price on tth day
c_l	Left end point centroid of IT2FS
c_r	Right end point centroid of IT2FS
c	Centroid of an IT2FS
c′	Measured value of $c(t)$ in centroid calculation
c^x	Type-1 centroid of $\mu_{A_x}(c(t))$
m_A	Mean values of the distributions of RMSE obtained by algorithms A
P_i	*I*th partition for close price time series (of MFTS)
Q_i	*I*th partition for variation series (of MFVS/SFVS/CSVS)
s_A	Standard deviation of the respective samples obtained by algorithms A
$V_M^d(t)$	Main factor variation series with delay d
$V_{S^i}^d$	*I*th secondary factor variation series with delay d
$V_S^d(t)$	Composite secondary variation series with delay d
W_{S^i}	Weight for ith secondary factor
$\mu_A(x)$	Type-1 membership function of linguistic variable x in fuzzy set A
$\bar{\mu}_{\tilde{A}}(x)$	Upper membership function of IT2FS $\tilde{A}$
$\underline{\mu}_{\tilde{A}}(x)$	Lower membership function of IT2FS $\tilde{A}$
Δ_{S^i}	Total difference variation for CSVS of ith secondary factor
$\hat{\Delta}_{S^i}$	Normalized value of Δ_{S^i} for CSVS of ith secondary factor

2.1 Introduction

Prediction of a time-series [1] refers to determining the amplitude of the series at time $t + 1$ from its previous m sample values located at time: $t, t - 1, t - 2, \ldots, t - (m - 1)$ for a finite positive integer m. An m-th order time-series prediction involves all the m previous sample values directly for its forecasting/prediction [2, 3]. In this chapter, we, for the sake of simplicity, however, use a first order prediction of time-series, where the $(t + 1)$-th sample of the time-series directly depends only on the sample value at time $(t + 1 - d)$, where d denotes the time-delay, although all the previous m sample values are required to design the prediction rules. There exists a vast literature on prediction of time-series for real processes, including rainfall [4, 5], population growth [6], atmospheric temperature [7], university enrollment for students [8–11], economic growth [12] and the like. This chapter is concerned with stock index, the time-series of which describing close price [13], is characterized by the following four attributes: non-linear [14], non-deterministic, non-stationary [15] and non-Gaussian jointly.

Designing a suitable model for stock index prediction requires handling the above four characteristics jointly. Although there exist several attempts to model time-series using non-linear oscillators [16], non-linear regression [17], adaptive auto-regression [18], Hzorth parameters [19] and the like, none of these could accurately model these time-series [20] for their inherent limitations to capture all the four characteristics jointly.

The logic of fuzzy sets plays a promising role to handle the above problems jointly. First, the nonlinearity of time-series is modeled by the nonlinearity of membership functions and their nonlinear mapping from antecedent to consequent space of fuzzy production rules. Second, the non-deterministic characteristics of the time-series (that might occur due to randomness in a wide space), is here significantly reduced because of its occurrence in one of a few equal sized partitions of the universe of discourse. Third, the non-stationary characteristics of the time-series that offers a correlation of signal frequencies with time [15] is avoided in fuzzy modeling by time-invariant models of membership functions [8]. Lastly, the non-Gaussian feature may be relaxed as locally Gaussian within small intervals (partitions) of the time-series. Thus, fuzzy sets are capable of capturing the uncertainty/imprecision in time-series prediction that might arise because of the above four hindrances.

The inherent power of fuzzy sets to model uncertainty of time-series has attracted researchers to employ fuzzy logic in time-series prediction. Song et al. [8–11] pioneered the art of fuzzy time-series prediction by representing the time-series value at time $t - 1$ and time t as fuzzy membership functions (MFs) and connected them by fuzzy implication relations for all possible time t in the time-series. If there exist n possible discrete values of time t, then we would have $n - 1$ possible fuzzy implication relations. Song et al. combined all these implication relations into a single relation R by taking union of all of these relations. The prediction involves first fuzzifying the crisp value of the time series at time t and then using composition rule of inference to determine the MF of the predicted time series at time $t + 1$ using R as the composite time-invariant implication relation. Lastly, they defuzzified the result to obtain the crisp value of the time-series at time $t + 1$.

The fundamental deviation in the subsequent work by Chen [21] lies in grouping of rules having common antecedents. Thus during the prediction phase, only few rules whose antecedent match with the antecedent of the fuzzified time-series value at time t only, need to be fired to obtain multiple MFs of the inferred consequences, one for each fired rule, an averaging type of defuzzification of which yields the actual inference at time $t + 1$. Hwang et al. considered a variation time-series [22] by taking the difference of two consecutive values of the time-series, and used max-product compositional rule of inference to predict the inference of the variation at time $t + 1$ from its previous values. A weighted average type of defuzzification was used to obtain the predicted value of the time-series at time $t + 1$. Cai et al. [23] introduced genetic algorithm to determine the optimal weight matrix for transitions of partitions of a given time-series from each day to its next day, and used the weight matrix to predict the time-series at time $t + 1$ from its value at time t. In [7], Chen et al. extended the work of Hwang et al. by first introducing a concept of secondary factors in the prediction of main factor time-series. There exists a vast literature on time-series prediction using fuzzy logic. A few of these that deserve special mention includes adaptive time-variant modeling [24], adaptive expectation modeling [25], Fibonacci sequence [26], Neural networks [27, 28], Particle Swarm Optimization [29] based modeling, fuzzy cognitive maps and fuzzy clustering [30], bi-variate [31, 32] and multi-variate [33–37] modeling and High order fuzzy multi-period adaptation model [38] for time-series prediction.

Most of the traditional works on stock index prediction developed with fuzzy logic [39] employ type-1(T1) fuzzy reasoning to predict future stock indices. Although T1 fuzzy sets have proved their excellence in automated reasoning for problems of diverse domains, including fuzzy washing machines [40, 41], fuzzy color TV [42] etc., they have limited power to capture the uncertainty of the real world problems [43]. Naturally, T1 fuzzy logic is incompetent to stock (and general time-series) prediction problems. The importance of interval type-2 fuzzy set (IT2FS) over its type-1 counterpart in chaotic time-series prediction has already been demonstrated by Karnik and Mendel [44]. There exist a few recent works attempting to model stock prediction problem using type-2 fuzzy sets [45, 46]. These models aimat representing a single (interval) type-2 membership function (MF), considering three distinct stock data items, called close, high and low prices [13]. Here too, the authors partitioned each of the above three time-series into intervals of equal size, and represented each partition as T1 fuzzy set. They constructed fuzzy If-Then rules describing transitions of stock index price from one day to the next day for each of the above time series. During prediction, they identified a set of rules containing antecedent fuzzy sets corresponding to current stock prices, obtained union and intersection of the consequents of the rules to derive (interval) type-2 fuzzy inferences and employed centre average defuzzifiers to predict the stock price for the next day. Bagestani and Zare [46] extended the above work by adaptation of the structure of the membership functions and weights of the defuzzified outputs to optimize root mean square error. In addition, the latter work employed centre of gravity defuzzifier in place of centre average defuzzifier used previously. The present chapter is an extension of the seminal work of Chen et al. [47] by the following counts.

1. In order to represent the close price $c(t)$ within a partition (interval), we represent each short duration contiguous fluctuation of $c(t)$ in a given partition of the universe of $c(t)$ by a type-1 MF, and take union of all these type-1 MFs within a partition to represent it by an interval type-2 fuzzy set (IT2FS). Under special circumstances, when a partition includes one or a few contiguous data points only, we represent the partition by a type-1 MF only.
2. The antecedent and consequent of fuzzy prediction rules of the form $A_i \rightarrow A_j$ (extracted from the consecutive occurrence of data points in partitions P_i and P_j, are represented by interval type-2 (IT2) (or type-1) fuzzy sets depending on the count and consecutive occurrences of data points in a partition. Naturally, there exist four possible types of fuzzy prediction rules: IT2 to IT2, IT2 to type-1, type-1 to IT2 and type-1 to type-1 depending on the representation of A_i and A_j by IT2 or type-1 fuzzy sets. This chapter thus employs four possible types of reasoning, each one for one specific type of prediction rule.
3. Appropriate defuzzification techniques, such as Karnik-Mendel algorithm for IT2 inferences [48, 49] and centroidal defuzzification for type-1 inferences have been employed to obtain the predicted close price at day $t + 1$.
4. Existing works [47] presume that the variation in secondary factor of the current day (of reliable reference time-series) identically influences the main factor of the next day. Naturally, if the interval counts in both the secondary factor and the main factor are equal, then the above variations have the same interval label in their respective universes. In the present chapter, we relax the restriction by considering all possible occurrence of variation of the main factor intervals for each occurrence of secondary factor interval obtained from the historical data. Such relaxation keeps the prediction process free from bias. In case, the current occurrence of secondary factor interval has no precedence in the historical variation data, we adopt the same principle used in [47].
5. One additional feature that caused significant improvement in performance in prediction is due to the introduction of evolutionary adaptation in the parameters of the selected structure of membership functions. The evolutionary adaptation tunes the base-width of the triangular/Gaussian membership functions (MFs) employed to reduce the root mean square error (RMSE) [27, 36]. Experiments undertaken reveal that tuning of parameters of MFs result in over 15% improvement in RMSE.

The proposed extensions introduced above outperforms all existing works on IT2 [45, 46] and type-1 fuzzy logic based stock index prediction techniques [8–11, 21, 47, 50] using RMSE as the metric.

The rest of this chapter is divided into five sections. In Sect. 2.2, we provide the necessary definitions required to understand this chapter. In Sect. 2.3, we present both training and prediction algorithms using IT2 fuzzy reasoning in the context of stock price prediction. Section 2.4 is concerned with experimental issues and computer simulation with details of results obtained and their interpretation. Performance analysis of the proposed technique with existing works is compared in Sect. 2.5. Conclusions are listed in Sect. 2.6.

2.2 Preliminaries

This section provides a few fundamental definitions pertaining to both time-series prediction and IT2FS. These definitions will be used in the rest of this chapter.

Definition 2.1 The last traded price in a trading day of a stock index is called *close price*, hereafter denoted by $c(t)$.

Definition 2.2 The stock index under consideration for prediction of a time series is called *Main Factor Time Series* (MFTS). Here, we consider TAIEX (Taiwan Stock Exchange Index) as the MFTS.

Definition 2.3 The associated indices of time series that largely influence prediction of the MFTS is called *Secondary Factor Time Series* (SFTS). Here, we consider NASDAQ (National Association of Securities and Dealers Automated Quotations) and DJI (Dow Jones Industrial average) as the SFTS.

Definition 2.4 For a given close price time series (CTS) $c(t)$, the *Variation Time Series* (VTS) [47] *with delay of d days* for close price is given by,

$$VTS^d(t) = \frac{c(t) - c(t-d)}{c(t-d)} \times 100 \tag{2.1}$$

for $t \in [t_{\min}, t_{\max}]$, where t_{min} and t_{max} denote the beginning and terminating days of the *training period* [47]. Here we consider $V_M^d(t)$ and $V_S^d(t)$ as the VTS for MFTS and SFTS respectively.

Definition 2.5 Prediction of MFTS $c(t+d)$ here refers to determining $c(t+d)$ from its historical values: $c(t)$, $c(t-1)$, $c(t-2)$, $c(t-3)$, …, $c(t-(m-1))$ and secondary factor VTS (SFVTS) $V_S^d(t)$, $V_S^d(t-1)$, …, $V_S^d(t-(m-1))$ for some positive integer m. Such prediction is referred to as m-th order forecasting. However, in most of the applications, researchers take $d = 1$ for simplicity and convenience [8–11, 47].

Definition 2.6 A T1 fuzzy set is a two tuple given by $<\mathrm{x}, \mu_{\mathrm{A}}(\mathrm{x})>$ where x is a linguistic variable in a Universe X and $\mu_{\mathrm{A}}(\mathrm{x})$ is the membership function of x in fuzzy set A, where $\mu_A(x) \in [0, 1]$.

Definition 2.7 A general type-2 Fuzzy Set (GT2FS) is a three tuple given by $<\mathrm{x}, \mu_{\mathrm{A}}(\mathrm{x}), \mu_{\mathrm{A}}(\mathrm{x}, \mu_{\mathrm{A}}(\mathrm{x}))>$ where x and $\mu_{\mathrm{A}}(\mathrm{x})$ have the same meaning as in Definition 2.6, and $\mu(\mathrm{x}, \mu_{\mathrm{A}}(\mathrm{x}))$ is the secondary membership in [0, 1] at a given $(x, \mu_{\mathrm{A}}(\mathrm{x}))$.

Definition 2.8 An interval type-2 fuzzy set (IT2FS) is defined by two T1 membership functions (MFs), called *Upper Membership Function (UMF)*, and *Lower Membership Function (LMF)*. An IT2FS $\tilde{A}$, therefore, is represented by $<\underline{\mu}_{\tilde{A}}(x), \bar{\mu}_{\tilde{A}}(x)>$ where $\underline{\mu}_{\tilde{A}}(x)$ and $\bar{\mu}_{\tilde{A}}(x)$ denote the lower and upper membership functions respectively. The secondary membership $\mu(\mathrm{x},\mu_{\mathrm{A}}(\mathrm{x}))$ in IT2FS is considered as 1 for all x and $\mu_{\mathrm{A}}(\mathrm{x})$.

Definition 2.9 The left end point centroid is the smallest of all possible centroids (of the embedded fuzzy sets [48]) in an IT2FS $\tilde{A}$ and is evaluated by

$$c_l = \frac{\sum_{i=1}^{k-1} \bar{\mu}_{\tilde{A}}(x_i) \cdot x_i + \sum_{i=k+1}^{N} \underline{\mu}_{\tilde{A}}(x_i) \cdot x_i}{\sum_{i=1}^{k-1} \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)} \tag{2.2}$$

using the well-known Karnik-Mendel algorithm [51],

where $x \in \{x_1, x_2, \ldots, x_N\}$ and $x_{i+1} > x_i \; \forall i = 1 \text{ to } N-1$. Here $x = x_k$ is a switch point and N denotes the number of sample points of $\bar{\mu}_{\tilde{A}}(x_i)$ and $\underline{\mu}_{\tilde{A}}(x_i)$.

Definition 2.10 The right end point centroid is the largest of all possible centroids (of the embedded fuzzy sets [48]) in an IT2FS $\tilde{A}$ and is evaluated by

$$c_r = \frac{\sum_{i=1}^{k-1} \underline{\mu}_{\tilde{A}}(x_i) \cdot x_i + \sum_{i=k+1}^{N} \bar{\mu}_{\tilde{A}}(x_i) \cdot x_i}{\sum_{i=1}^{k-1} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \bar{\mu}_{\tilde{A}}(x_i)} \tag{2.3}$$

using the well-known Karnik-Mendel algorithm [51], where $x \in \{x_1, x_2, \ldots, x_N\}$ and $x_{i+1} > x_i \; \forall i = 1 \text{ to } N-1$. Here $x = x_k$ is a switch point and N denotes the number of sample points of $\bar{\mu}_{\tilde{A}}(x_i)$ and $\underline{\mu}_{\tilde{A}}(x_i)$.

Definition 2.11 The centroid of an IT2FS is given by

$$c = \frac{(c_l + c_r)}{2} \tag{2.4}$$

where c_l and c_r are the left and the right end point centroids.

2.3 Proposed Approach

Given a time-series $c(t)$ for close price of a stock index, we observe consecutive 10 months' daily data for the above time-series to extract certain knowledge for prediction of the time series. To extract such knowledge, we partition the entire range of $c(t)$ into equal sized intervals P_i, $i = 1$ to p, and determine the list of possible changes in $c(t)$ from day $t = t_i$ to $t = t_{i+d}$ for any valid integer i and a fixed delay d. Classical production rule-based reasoning [52] could be performed to predict the interval of $c(t_{i+d})$ from the known interval of $c(t_i)$ using the previously acquired rules. However, because of uncertainties in time-series, the strict production rules may not return the correct predictions. The logic of fuzzy sets, which has proved itself a successful tool to handle uncertainty, has therefore been used here to predict the membership of $c(t_{i+d})$ in a given partition P_{i+d} from the measured membership of $c(t_i)$ in partition P_i.

In this chapter, each continuum neighborhood of data points of $c(t)$ in a given partition P_i is represented by a T1 fuzzy set, and the union of all such T1 fuzzy sets under the partition is described by an IT2FS. The IT2FS model proposed for each individual partition can capture the uncertainty of the disjoint sets of data points within the partition. In addition, the transition: $c(t)$ to $c(t+d)$ from partition P_i to partition P_j is encoded as an IT2 fuzzy prediction rule, rather than a typical binary production rule. The IT2 prediction rule indicates that the linguistic variables present in the antecedent and consequent parts of the rule have IT2 MFs. The prediction of $c(t'+d)$ from a given measurement point $c(t')$, is done in two steps. In the first step, we use fuzzy reasoning to determine the membership of $c(t'+d)$ in partition P'_j from the known membership of $c(t')$ in P'_i. After the inference is obtained, we use a T1/IT2 de-fuzzification depending on the type of reasoning used. The modality selection of reasoning (i.e., T1/IT2) is performed based on the distribution of data points in a given partition. This is undertaken in detail in the algorithm to be developed for rule identification from transitions history of data points in the time-series.

The principle of time-series forecasting introduced above is expected to offer good prediction accuracy, in case the time-series under consideration (called main factor) is not disturbed by external influences, such as changes in Government policies, macro/micro economic conditions, and many other unaccountable circumstances. Since all the external influences are not known, in many circumstances we model the influences by considering variation from other associated (secondary) world indices. Chen et al. [47] introduced an innovative approach to represent the effect of secondary indices to the main factor time-series. They considered composite variation of several secondary indices by measuring the deviation of individual index from the main factor time-series, and later used these deviations to determine normalized weights. These normalized weights are used later to scale the stock indices to determine the composite variation of secondary stock indices. To predict a stock data at day $t+1$ from the measurements of the same stock data at day t, Chen et al. determined the partition of the composite variation at day t with an assumption that the main factor at day t too would have the same partition. Later they used T1 fuzzy reasoning (using acquired rules in the training phase) to predict the stock data for the main factor at day $t+1$.

This chapter proposes three alternative approaches to economic time-series prediction. The first proposal considers employing IT2FS in place of T1 fuzzy reasoning introduced in [8–11, 21, 22, 47]. The IT2FS captures the inherent uncertainty in the time-series and thus provides a better fuzzy relational mapping from the measurement space to inference space, thereby offering better performance in prediction than its T1 counterpart. The second approach considers both IT2FS based reasoning along with feed-forward information from secondary stock indices, which usually are of more relative stability than the time-series under prediction. Thus the performance with feed-forward from secondary time-series gives better relative performance in comparison to the only IT2FS based reasoning. The third approach considers adaptation of T1 membership functions used to construct IT2FS MFs along with feed-forward connections from secondary stock indices. The performance of the third approach is better than its other two counterparts. The main steps of the algorithms are outlined below.

2.3.1 Training Phase

Given the MFTS and the SFTS of close price $c(t)$ for 10 months, we need to determine (i) group of type-2 fuzzy logical implications (prediction rules) for individual interval of main factor variation, and (ii) Secondary to Main Factor variation mapping. This is done by the following six steps.

1. Partitioning of main factor close price (MFCP) into p intervals (partitions) of equal length.
2. Construction of IT2 or T1 fuzzy sets as appropriate for each interval of close price.
3. IT2 or T1 fuzzy rule base selection for each interval.
4. Grouping of IT2/T1 fuzzy implication for individual main factor variation time-series $V_M^d(t)$.
5. Computing Composite Secondary Variation Series (CSVS) and its partitioning.
6. Determining secondary to main factor variation mapping.

Figures 2.1, 2.2, 2.3, 2.4, 2.5, 2.6 and 2.7 together explains the steps involved in the training phase. The details of individual steps are given below point-wise.

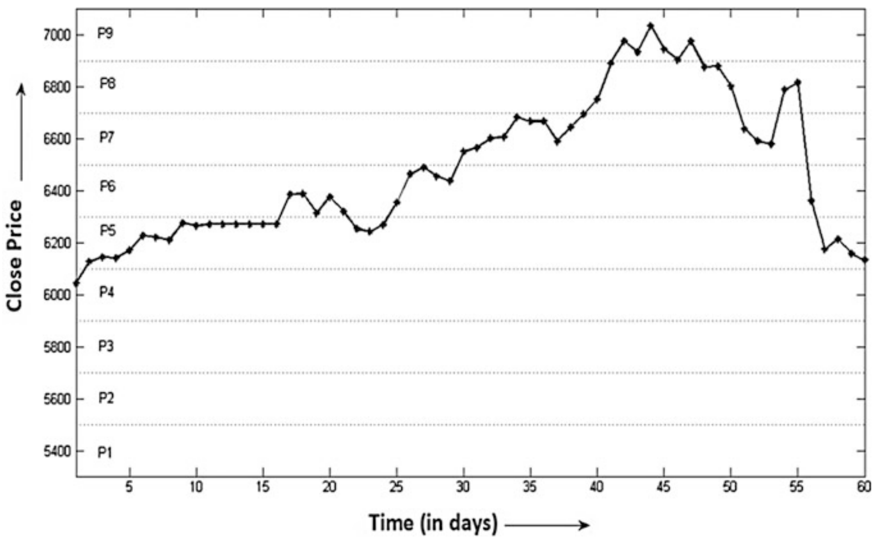


Fig. 2.1 Time Series $c(t)$ and the partitions

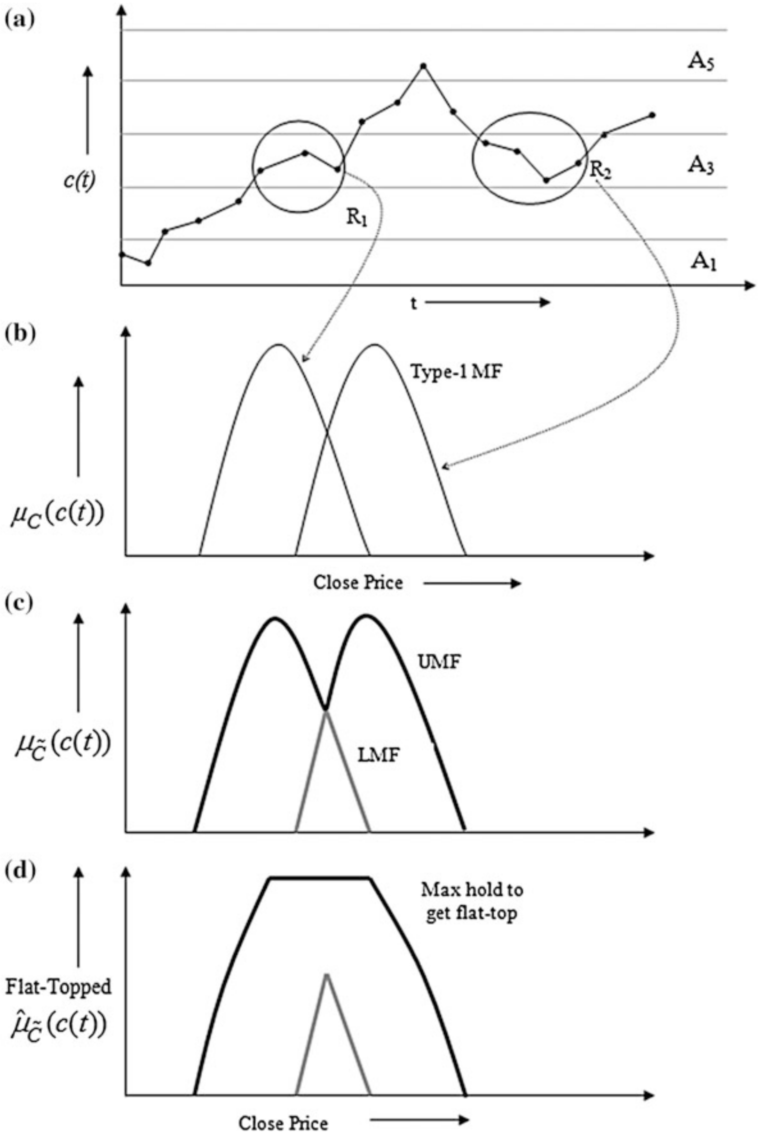


Fig. 2.2 Construction of flat-top IT2FS for partition A3: **a** The close price, **b** Type-1 MFs for regions R1 and R2, **c** IT2FS representation of (**b**), **d** Flat-top IT2FS obtained by joining the peaks of two lobes

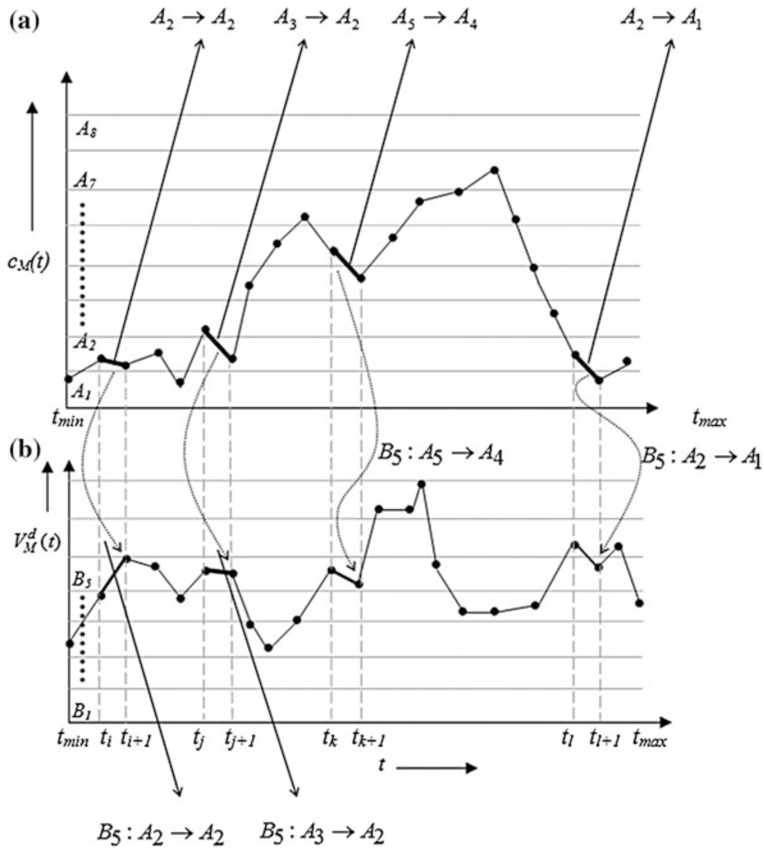


Fig. 2.3 Construction of fuzzy logical implication and their grouping under MFVS $V_M^d(t)$ with d = 1: **a** If $c_M(t_i) \in A_k$ and $c_M(t_i + 1) \in A_j$; then the rule is $A_k \rightarrow A_j$, **b** If $V_M^d(t_i + 1) = B_s$, $\exists s$, then grouping is done as B_s:$A_k \rightarrow A_j$, $\exists j, k, s$

2.3.1.1 Partitioning of Main Factor Close Prices into p Intervals of Equal Length

Consider a universe of discourse U given by $[MIN - D_1,\ MAX + D_2]$, where MAX and MIN are the respective global maximum and global minimum of the time-series for close price $c(t)$ for a given duration of t in $[1,\ 10]$ months. D_1 and D_2 are positive real numbers in $[1,\ 99]$, such that $(MAX + D_1)/100$ and $(MIN - D_2)/100$ are positive integers. Divide the universe U into p disjoint partitions: $P_1, P_2, \ldots, P_p$ of equal intervals as given in Fig. 2.1 [53] for more precision and clarity), where the length of an interval [47] is given by $[(MAX + D_1) - (MIN - D_2)]/p$.

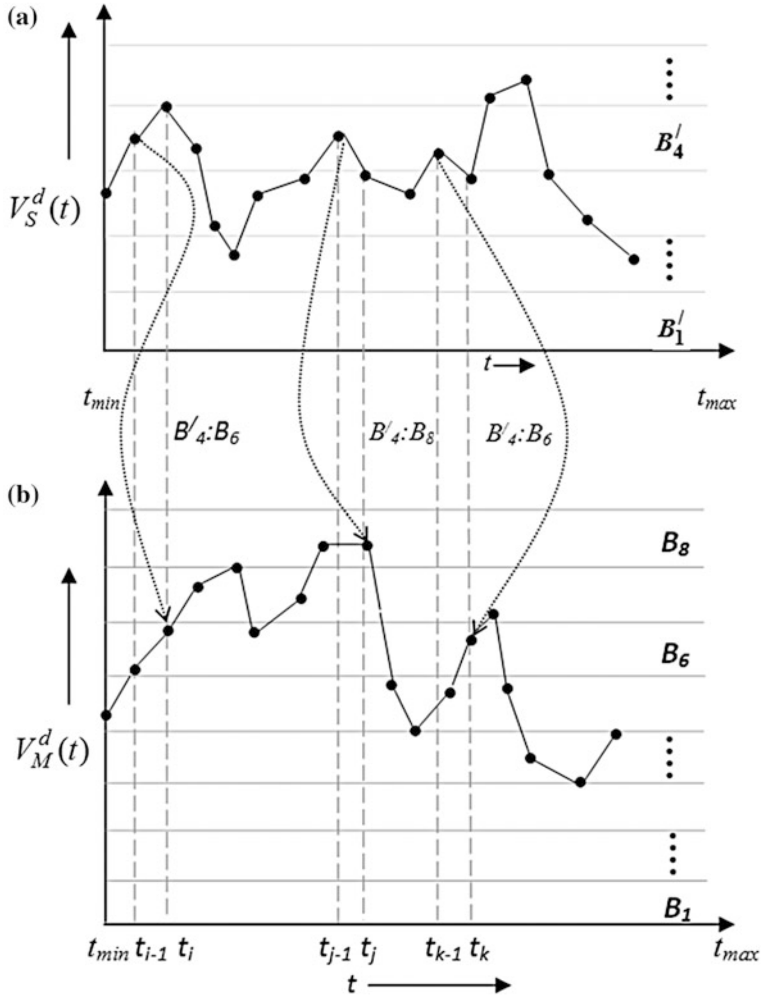


Fig. 2.4 Secondary to Main factor variation mapping considering d = 1: If $V_M^d(t) \in B_k$, then the mapping is written as $B'_j : B_k$

2.3.1.2 Construction of IT2 or Type-1 Fuzzy Sets as Appropriate for Each Interval of Close Price

For each partition P_i, $i = 1$ to p of $c(t)$, and for each set j of consecutive data points in P_i, we define fuzzy sets $A_{i,\,j}$ for $j = 1$ to j_{Max}, where the T1 MF of $A_{i,j}$ indicates the linguistic membership function (MF) CLOSE_TO_CENTRE_VALUE. For each group j of (three or more) consecutive data points of $c(t)$ in P_i, construct a

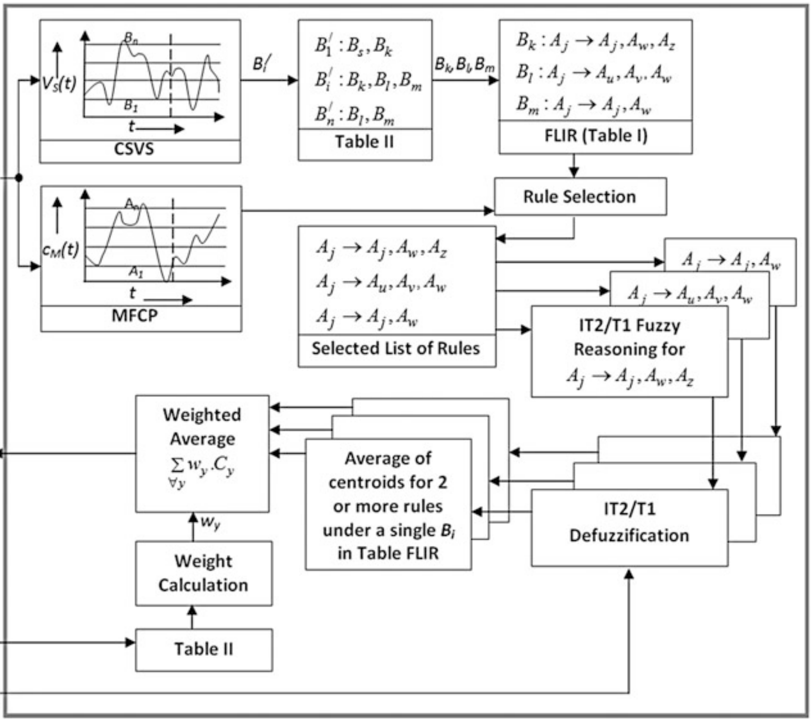


Fig. 2.5 The main steps in the prediction algorithm of a stock index time-series considering d = 1

Gaussian T1 membership function with mean and standard deviation equal to the respective mean and standard deviation of these data points. Construct an IT2FS $\tilde{A}_i$, the footprint of uncertainty FOU_i of which is obtained by taking the union of $A_{i,j}$ for $j = 1$ to j_{Max}. The constructed FOU is approximated (by joining the peaks of T1 MFs with a straight line of zero slope) with a flat top UMF to ensure convexity and normality [54, 55] of the IT2FS. The following special cases need to be handled for partitions with fewer data points.

If a partition P_i includes only one data point of $c(t)$, we construct a T1 Gaussian MF with mean equal to the data point and very small variance of the order of 10^{-4} or smaller. If a partition P_i includes only two consecutive data points of $c(t)$ we construct a T1 Gaussian MF with mean and standard deviation equal to the respective mean and standard deviation of these two data points. Lastly, if a partition P_i includes only two (or more) discrete individual data points of $c(t)$ we construct two (or more) Gaussian MFs with means equal to the respective data points and very small variance of the order of 10^{-4} or smaller. We now construct a IT2FS by taking union of these T1 MFs.

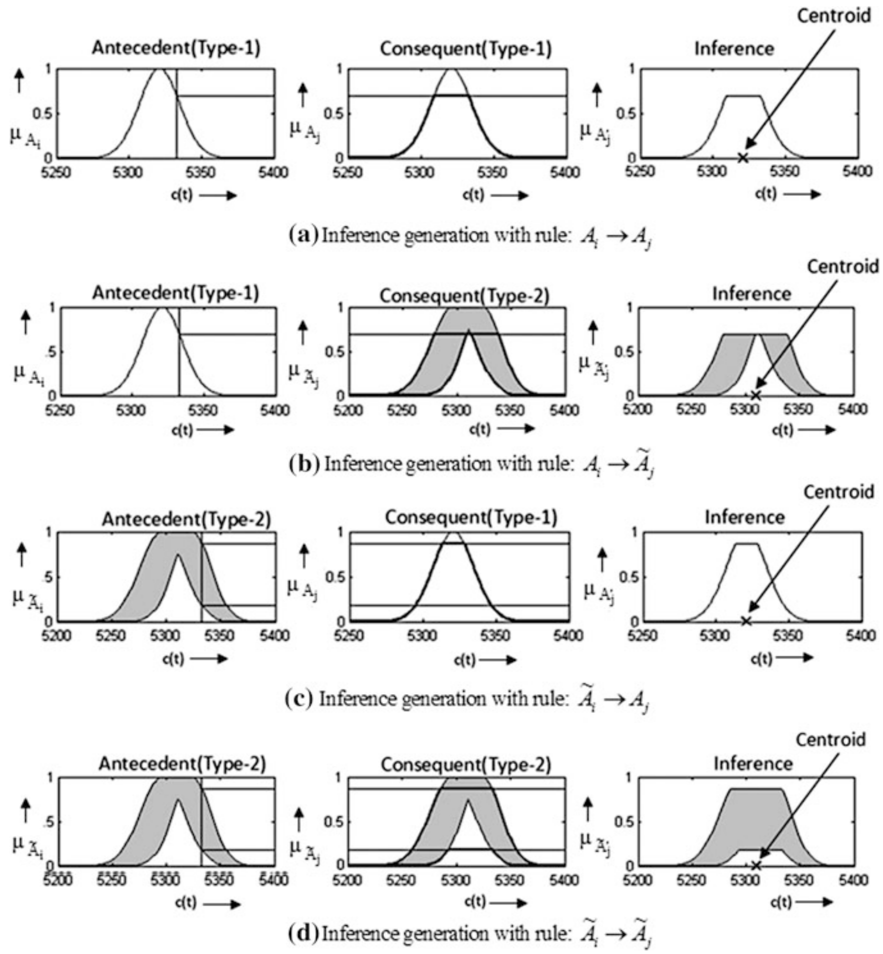


Fig. 2.6 Inference Generation with T1/ IT2 antecedent-consequent pairs

2.3.1.3 Fuzzy Prediction Rule (FPR) Construction for Consecutive $c(t)$s

For each pair of training days t and $t + d$, we determine the mapping from $\tilde{A}_i$ to $\tilde{A}_j$, where $\tilde{A}_i$ and $\tilde{A}_j$ correspond to IT2 MF of the close prices at day t and day $t + d$ respectively.

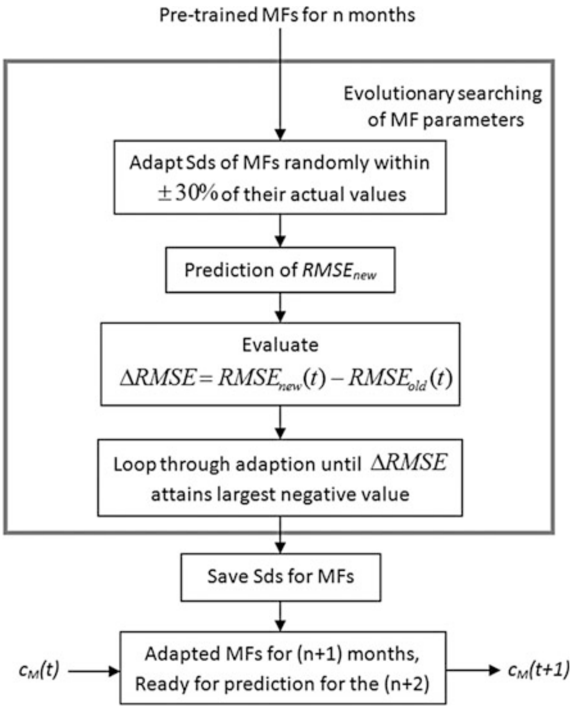


Fig. 2.7 Optimal Selection of MFs to minimize RMSE

2.3.1.4 Grouping of IT2/T1 Fuzzy Implications for Individual Main Factor Variation $V_M^d(t)$

(a) **MFVS Construction**: For trading days $t \in [t_{Min}, t_{Max}]$, we evaluate $V_M^d(t)$ using Eq. (2.1) for the main factor (here, TAIEX [56]).

(b) **Partitioning of $V_M^d(t)$ into B_is**: Although in most of the practical cases $V_M^d(t)$ lies in $[-6\%, +6\%]$, we here consider a wider range of it in $[-\infty, +\infty]$, so as to not to disregard the possibility of occurrences of stray data points outside $[-6\%, +6\%]$. Partitioning the entire space of $[-\infty, +\infty]$ is performed by segregating the range: $[-6\%, +6\%]$ into equal sized intervals and the range beyond on either sides of it, i.e., $[-\infty, -6\%)$ and $(+6\%, +\infty]$ into two distinct intervals. Such partitioning ensures a uniformly high probability of occurrence of any data point at any one of the intervals for the band $[-6\%, +6\%]$ and a uniformly low probability of occurrence to any data point lying in $[(-\infty, -6\%)$ and $(+6\%, +\infty)]$ ranges. The entire space of $V_M^d(t)$ in $[-\infty, +\infty]$ is divided into 14 intervals (partitions) B_1 through B_{14}, where interval B_1 describes the range $[(-\infty, -6\%)$, B_2, through B_{13} represent 12 partitions covering $[-6\%, +6\%]$ in order of increasing values of $V_M^d(t)$, and the interval B_{14} represents the last partition $(+6\%, +\infty]]$.

Table 2.1 Main factor fuzzy logical implication (FLI) considering d = 1

Group	Time points	
	t_1 … t_{i+1}	t_{j+1} … t_{k+1} … t_{l+1} … t_n
B_1 …	… …	… …
B_5 …	$A_2 \to A_2$ …	$A_3 \to A_2 A_5 \to A_4 A_2 \to A_1$ …
B_{14}	…	…

Table 2.2 Main factor fuzzy logical implication (FLI) under variation groups considering d = 1

Group	Antecedent of main factor				
	A_1	A_2	A_3 …	A_5 …	A_{18}
B_1	…	…	…	…	
…	…	…	…	…	…
B_5		$A_2 \to A_2$ $A_2 \to A_1$	$A_3 \to A_2$	$A_5 \to A_4$	
…	…	…	…	…	…
B_{14}	…	…	…	…	…

(c) **Grouping of FPRs Under Each Variation Group** B_i: For each feasible $t+d$ in $[t_{Min}, t_{Max}]$, find the partition B_i, such that $V_M^d(t+d)$ lies in the range of B_i. Also obtain the fuzzy sets $\tilde{A}_j, \tilde{A}_k$ corresponding to the partitions P_j and P_k at days t and $t+d$ respectively. Then construct a rule: $\tilde{A}_j \to \tilde{A}_k$ with a label B_i, represented by

$$B_i : \tilde{A}_j \to \tilde{A}_k.$$

Repeat this step $\forall feasible\, t \in [t_{Min}, t_{Max}]$. Figure 2.3 describes the above mapping of $\tilde{A}_j \to \tilde{A}_k$ and its labeling against B_i for a MFTS $c_M(t)$. Tables 2.1 and 2.2 clarifies the grouping of rules like $\tilde{A}_j \to \tilde{A}_k$ in B_i following Fig. 2.3.

2.3.1.5 Computing Composite Secondary Variation Series (CSVS) and Its Partitioning

(a) **Computing Secondary Factor Variation Series (SFVS)**: For ith elementary secondary factor SF^i, we evaluate $V_{S^i}^d(t)$ (variation in SF^i) using Eq. (2.1), where $c^i(t)$ and $c^i(t-d)$ denote the close price of tth day and $(t-d)$th day of i-th elementary secondary factor respectively.
(b) **Total Difference Variation** (Δ_{s^i}) **Computation**: For i-th elementary secondary SF^i, evaluate total difference variation, denoted by Δ_{s^i} by using the following expression,

$$\Delta_{s^i} = \sum_{\forall t} |V_{S^i}^d(t-d) - V_M^d(t)| \tag{2.5}$$

where $V_M^d(t)$ and $V_{S^i}^d(t-d)$ denote the variation is main factor at day t and that in ith SF^i day $(t-d)$.

(c) **Normalization**: Use transformation (6) to obtain the normalized value of Δ_{s^i}.

$$\hat{\Delta}_{s^i} = \left[\frac{\Delta_{s^i}}{\sum_{j=1}^{n} \Delta_{s^j}}\right]^{-1} = \frac{\sum_{j=1}^{n} \Delta_{s^j}}{\Delta_{s^i}} \tag{2.6}$$

where index j in $[1, n]$ refer to different elementary SF^{i}s.

(d) **Weight Computation**: Determine the normalized weighted variation for elementary SF over the training period (January 1 through October 31 of any calendar year).

$$W_S^i = \frac{\hat{\Delta}_{s^i}}{\sum_{j=1}^{n} \hat{\Delta}_{s^j}} \tag{2.7}$$

(e) **Composite Secondary Variation Series (CSVS) Computation** [47]: The overall variation at day t is given by

$$V_S^d(t) = \sum_{i=1}^{n} V_{s^i}^t \cdot W_{s^i}^t \tag{2.8}$$

2.3.1.6 Determining Secondary to Main Factor Variation Mapping

Like the main factor time-series, the secondary factor variation series $(CSVS)$ is also partitioned into 14 intervals: $B'_1, B'_2, \ldots., B'_{14}$ following the same principle as introduced in step 4(b). For each $V_S^d(t-d)$ lying in B'_i, and for each $V_M^d(t)$ lying in $B_j, B_k, \ldots, B_l$, for all t, we group $B_j, B_k, \ldots, B_l$ under group B'_i.

Group $B'_i : B_j,\ B_k,\ \ldots, B_l$

Figure 2.4 illustrates the principle of group formation under B'_i. Here, for space limitation we show only 8 intervals B_1 through B_8 instead of 14 intervals. The frequency count of B_j in MFTS at day t for a given B'_i in CSVS at day $t-d$ is evaluated in Fig. 2.4 and included in Table 2.3 for convenience.

Table 2.3 Frequency of occurrence of main factor variation in each group of secondary factor variation considering d = 1

From Secondary Factor Variation ↓ / To Main Factor Variation →	B_1	B_2	…	B_6	…	B_8	…	B_{14}
B_1'	…	…	…	…	…	…	…	…
B_2'	…	…	…	…	…	…	…	…
	…	…	…	…	…	…	…	…
B_4'	**0**	**0**	…	**2**	…	**1**	…	**0**
	…	…	…	…	…	…	…	…
B_{14}'	…	…	…	…	…	…	…	…

2.3.2 Prediction Phase

Prediction of time-series at day $t + d$ from its close price at day t could easily be evaluated by identifying all the rules having antecedent A_j, where A_j, denotes the fuzzy set corresponding to the partition at day t in the partitioned main factor close price time-series. However, it is observed by previous researchers [8–11] that prediction using all the rules with A_j as antecedent does not give good results. This chapter overcomes the above problem by selecting a subset of all possible rules with A_j as antecedent. The subset-hood is determined by using the secondary variation time series. For example, if the partition returned by secondary factor time-series at day t is B_i' then we obtain the corresponding partitions in MFTS by consulting the secondary to main factor variation series mapping introduced in Table 2.3. Suppose the Table 2.3 returns as the partitions in the MFTS. We now look for rules having antecedent A_j in the labels in Table 2.2. The rules present under a given label are fired , and the average of the centroids of the resulting inferences is preserved. The weighed sum of the preserved average centroids (corresponding to individual labels) is declared as the predicted close price at day $t + d$. The algorithm for close price prediction is given below. Figure 2.5 provides the steps of the prediction algorithm schematically.

1. Obtain secondary variation B_i' and main factor close price A_j both for day t.
2. Using Table 2.3, determine B_k, B_l, B_m etc. of main factor variation enlisted against B_i'.

3.

(a) **Rule Selection**: Identify fuzzy production rules with antecedent A_j against rules with main factor variation B_k, B_l, B_m in Table 2.2.

(b) For each production rule under a given B_p, $p \in \{k, l, m\}$.,

i. **IT2/T1 Fuzzy Reasoning**: Perform IT2/T1 fuzzy reasoning with rules $A_j \to A_u$, $A_j \to A_v$, $A_j \to A_w$ if the row B_p in Table 2.2 includes the rule $A_j \to A_u, A_v, A_w$. If the group B_p does not contain any rule then we consider the mid value of the partition corresponding to the partition A_j for forecasting following [47].

ii. **IT2/T1 Defuzzification:** Employ IT2 or T1 defuzzification, as applicable, to obtain centroids of the discretized MFs: A_u, A_v, A_w and take the average of the centroids.

The procedure of reasoning and defuzzification considering the presence of T1/IT2 MFs in antecedent/consequent is given separately for convenience of the readers.

(c) **Weight Calculation and Prediction**: Determine the frequency counts f_k, f_l, f_m of the main factor variation B_k, B_l, B_m under secondary variation B_i' in Table 2.3 to determine the probability of occurrences as $f_k/(f_k + f_l + f_m)$, $f_l/(f_k + f_l + f_m)$, $f_m/(f_k + f_l + f_m)$. We use the probabilities to defuzzify expected value of main factor close price for the next day by taking sum of products of probabilities and average centroids values under B_k, B_l, B_m.

The complete steps of prediction of a time series are illustrated in Fig. 2.5.

Procedure of T1/IT2 Reasoning and Defuzzification

Case I: When Both Antecedent and Consequents are IT2FS

(a) **IT2 Reasoning**: Let $\tilde{A}_i(c(t))$ be the value of $\tilde{A}_i$ for linguistic variable $x = c(t)$. Let UMF and LMF for $\tilde{A}_i$ be $UMF_i(c(t))$ and $LMF_i(c(t))$ respectively. On firing the rules: $\tilde{A}_i \to \tilde{A}_j$, $\tilde{A}_i \to \tilde{A}_k$ and $\tilde{A}_i \to \tilde{A}_l$ we determine the fuzzy inferences by the following procedure. For the rule: $\tilde{A}_i \to \tilde{A}_j$, the IT inference is obtained by

$$UMF_j' = Min\,[UMF_i(c'),\, UMF_j], \tag{2.9}$$

$$LMF_j' = Min\,[LMF_i(c'),\, LMF_j], \tag{2.10}$$

where c' is a measured value of $c(t)\,\exists t$.

Similarly, we obtain UMF_y' and LMF_y' by replacing index j by y for $y \in \{k, l\}$ for the remaining rules.

(b) **IT2 Centroid Computation**: For each pair of UMF'_x and LMF'_x for $x \in \{j, k, l\}$, we determine the centroid of IT2FS A'_x by the following method. Determine the lower End Point centroid c^x_L and Upper End Point centroid c^x_R, for $x \in \{j, k, l\}$ and centroid c^x following Eqs. 2.2, 2.3 and 2.4 respectively. Thus for the rule: $B_i : \tilde{A}_i \to \tilde{A}_j, \tilde{A}_k, \tilde{A}_l$, we obtain the c^j, c^k and c^l. Determine the average of c^j, c^k and c^l.

Case II: When Antecedent is T1FS and Consequent is IT2FS

(a) **IT2 Reasoning**: For the rule: $B_i : A_i \to \tilde{A}_j, \tilde{A}_k, \tilde{A}_l$ we determine the fuzzy inferences by the following procedure for

$$UMF'_j = Min\,[\mu_{A_i}(c'),\ UMF_j], \tag{2.11}$$

$$LMF'_j = Min\,[\mu_{A_i}(c'),\ LMF_j], \tag{2.12}$$

where c' is a measured value of $c(t)\,\exists t$ and $\mu_{A_i}(c')$ is the T1 membership at $c(t) = c'$. Similarly, we obtain UMF'_y and LMF'_y by replacing index j by y for $y \in \{k, l\}$.

(b) **IT2 Centroid Computation**: The centroid computation procedure, here, is similar to that in Case I.

Case III: When Antecedent is IT2FS and Consequent is T1 FS

(a) **T1 Reasoning**: For the rule: $B_i : \tilde{A}_i \to A_j, A_k, A_l$, we determine the fuzzy inferences by the following procedure.

$$\mu_{A'_j}(c(t)) = Min\,[UMF_i(c'), \mu_{A_j}(c(t))], \tag{2.13}$$

where c' is a measured value of $c(t)\,\exists t$.

Similarly, we obtain $\mu_{A'_y}(c(t))$ by replacing index j by y for $y \in \{k, l\}$.

(b) **T1 Centroid Computation**: For each T1 discretized MF $\mu_{A'_x}(c(t))$ for $x \in \{j, k, l\}$, we determine the T1 centroid of A'_x by the following formula

$$c^x = \frac{\sum_{c(t_i)=-\infty}^{\infty} \mu_{A_x}(c(t_i)).c(t_i)}{\sum_{c(t_i)=-\infty}^{\infty} \mu_{A_x}(c(t_i))} \quad \text{For } x \in \{j, k, l\} \tag{2.14}$$

Thus for the rule $B_i : \tilde{A}_i \to A_j, A_k, A_l$, we obtain the centroids c^j, c^k and c^l. Determine the average of c^j, c^k and c^l.

Case IV: Antecedent and Consequent Both are T1 FS

(a) **T1 Reasoning**: For the rule: $B_i : A_i \rightarrow A_j, A_k, A_l$ we determine the fuzzy inferences by the following procedure.

$$\mu_{A'_j}(c(t)) = Min\,[\mu_{A_i}(c'), \mu_{A_j}(c(t))],$$

Where c' is a measured value of $c(t)\,\exists t$.
Similarly, we obtain $\mu_{A'_y}(c(t))$ by replacing index j by y for $y \in \{k, l\}$.

(b) **T1 Centroid Computation**: The centroid computation procedure, here, is similar to that in Case III.

Figure 2.6 provides the inference generation mechanism introduced above graphically for the above four cases.

2.3.3 Prediction with Self-adaptive IT2/T1 MFs

Large scale experiments with time-series prediction reveal that the results in RMSE are highly influenced by the shape of MFs used in the antecedent/consequent of the prediction rules. This motivated us to arrange on-line selection of MFs from a standard list, here triangular and Gaussian, with provisions for variations in their base-width. The optimal selection of base width can be performed by employing an evolutionary algorithm with an ultimate aim to minimize the RMSE. Any standard evolutionary/swarm algorithm could serve the above purpose. However, for our experience of working with Differential Evolution (DE) algorithm [57, 58] coupled with its inherent merits of low computational overhead, simplicity, requirement of fewer control parameters and above all its high accuracy, we used DE to adaptively select the right structure of MFs with RMSE as the fitness function.

Figure 2.7 provides a schematic overview of the MF adaptation scheme. The bold box in Fig. 2.7 includes the complete adaptation steps, while the bottommost block represents the prediction algorithm with adapted parameters. The adaptation module makes trial selection of standard deviation (base-width) of the Gaussian (triangular) MFs within ±30% of their original values. Next the change in RMSE due to adoption of the new MFs is evaluated. Finally, we loop through the above steps until no further reduction in change in RMSE is observed. The last obtained values of parameters of MFs are saved for subsequent usage in prediction. The benefits of the adaptation of MFs is compared in the next section vide Fig. 2.8.

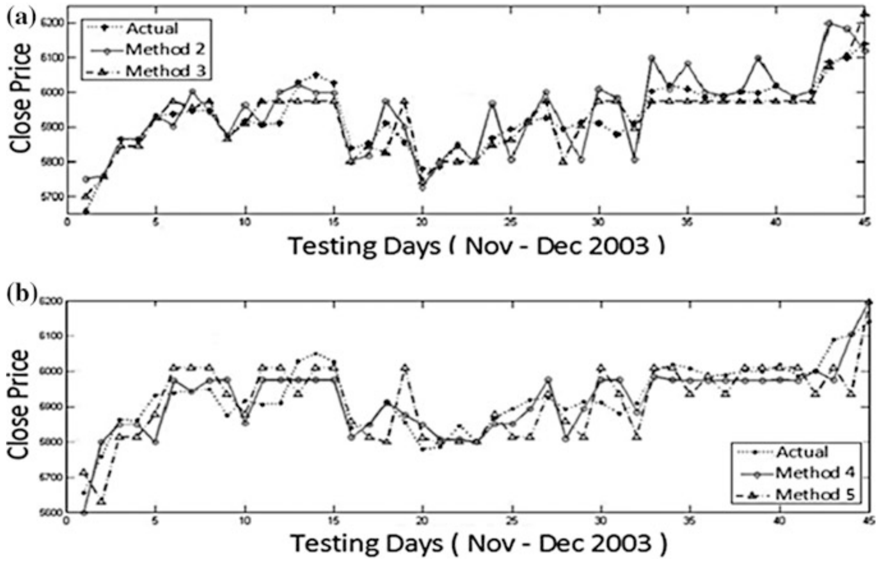


Fig. 2.8 Forecasted TAIEX of the months November and December 2003 using Gaussian MFs and Triangular MFs, **a** Actual TAIEX, forecasted data using proposed method 2 without adopting the Gaussian MFs and forecasted data using proposed method 3 adopting the Gaussian MFs, **b** Actual TAIEX, forecasted data using proposed method 4 without adopting the triangular MFs and forecasted data using proposed method 5

2.4 Experiments

The experiment includes both training and testing with TAIEX [56] close price [13], hereafter called main factor, and NASDAQ and DOWJONES close prices as secondary factors. The training session comprises T1/IT2 membership function construction, extraction of fuzzy prediction rules and mapping of secondary to main factor variations. The testing session comprises fuzzy prediction rule selection for firing, T1/IT2 fuzzy reasoning as applicable, defuzzification and weighted averaging of multiple defuzzified rules falling under different main factor variations. While performing the experiments, we consider five distinctive methods. The proposed method 1 includes only IT2FS based reasoning, ignoring the effect of secondary factor without adaptation of membership function is given only for the sake of academic interest. The proposed method 2 to proposed method 5 includes the influence of secondary factor. The proposed method 2 to proposed method 5 are hereafter called (a) Proposed method 2: fixed Gaussian MF (without adaptation), (b) Proposed method 3: Gaussian with provisions for adaptation in standard deviation, (c) Proposed method 4: fixed triangular MF (without adaptation), and (d) Proposed method 5: triangular with provisions for adaptation in base-width. The training and prediction algorithms incorporating the above five types of MFs are hereafter referred to as proposed methods: 1, 2, 3, 4 and 5 respectively for brevity.

Table 2.4 Strategies adopted in various experimental proposed methods

Methods	Reasoning	Membership function considered	Secondary factor considered	Adaptation
Proposed method 1	IT2	Gaussian	No	No
Proposed method 2	T1 and IT2 combined	Gaussian	Yes	No
Proposed method 3	T1 and IT2 combined	Gaussian	Yes	Yes
Proposed method 4	T1 and IT2 combined	Triangular	Yes	No
Proposed method 5	T1 and IT2 combined	Triangular	Yes	Yes

For clarity, we summarized the strategies adopted in these five methods are shown in Table 2.4. The initial MFs in Proposed method 4 and Proposed method 5 are represented by isosceles triangles with a peak membership of one at the centre and base-width equal to 6σ, where σ denotes the standard deviation of the consecutive data points in a given partition of close price.

2.4.1 Experimental Platform

The experiment was performed using MATLAB 2012b under WINDOWS-7 operating system running on a IBM personal computer with Core i5 processor with system clock of 3.60 G-Hz frequency and system RAM of 8 GB.

2.4.2 Experimental Modality and Results

2.4.2.1 Policies Adopted

The close price data for both main factor and secondary factors are obtained for the period 1990–2004 from the website [56]. The training session was fixed for 10 months: January 1 to October 31 of each year on all trading days. In case all the trading days of secondary factors do not coincide with those of the main factor, we adopt two policies for the following two cases. Let Set A and B denote the dates of trading in main and secondary factors respectively. If $A - B$ (A minus B) is a non-null set, then the close price of previous trading days in secondary factor has to be retained over the missing (subsequent) days. If $B - A$ is non-null set, then we adopt the following policies. First, if the main factor has missing trading days due to holidays and/or other local factors, then no training is undertaken on those days.

Second, in the trading of next day of main factor, we consider the influence of the last day of trading in secondary closing price. After the training is over, the following items including prediction rules (also called Fuzzy Logic Implications (FLI)) and secondary to main factor variation groups are saved for the subsequent prediction phase. The prediction was done for each trading day during the month of November and December. Comparison of the results of prediction with those of Chen et al. [47] is given in authors' webpage [53], and is not given here for space restriction. The results of prediction (November-December, 2003) with and without adaptation of parameters (standard deviations) of MFs are given in Fig. 2.8 along with the actual close price.

2.4.2.2 MF Selection

Experiments are performed with both Gaussian and triangular T1 MFs. The UMF (LMF) of the IT2FS is obtained by taking maximum (Minimum) of the T1 MFs describing the same linguistic concept obtained from different sources. Figure 2.2 respectively provides the construction of IT2FS from triangular and Gaussian T1 MFs, following the steps outlined in Sect. 2.3. The relative performance of triangular and Gaussian MFs is examined by evaluating RMSE of the predicted close price with respect to its actual TAIEX values. In most of the test cases, prediction of close price is undertaken during the months of November and December of any calendar year between 1999 and 2004.

The RMSE plots shown in Fig. 2.8 reveal that triangular MFs yield better prediction results (less RMSE) than its Gaussian counterpart. For example, the RMSE for TAIEX for the year 2003 using triangular and Gaussian MFs are respectively found to be 37.123 and 47.1108 respectively, justifying the importance of triangular MFs over Gaussian ones in the time-series prediction.

2.4.2.3 Adaptation Cycle

The training algorithm is run with the close price time-series data from January 1st to October 31st on all trading days. For tuning the T1 MFs (before IT2FS construction) for qualitative prediction, the adaption algorithm is run for the period of September 1st to October 31st for the subsequent prediction of November. After the prediction of November month is over, the adaption procedure is again repeated for the month of October 1st to November 30th in order to predict the TAIEX close price in December. Such adaption over two consecutive months is required to track any abnormal changes (such as excessive level shift) in the time-series.

The improvement in performance due to inclusion of adaptation cycles is introduced in Fig. 2.8 (see [53] for precision), obtained by considering Gaussian MFs. It is apparent from Fig. 2.8a that in presence of adaptation cycles, the RMSE appears to be 47.1108, while in absence of adaptation, RMSE is found to be 52.771. The changes in results (RMSE) in presence of adaptation cycles due to use of

Table 2.5 Comparison of RMSEs obtained by the proposed technique with existing techniques

Methods	Years						
	1999	2000	2001	2002	2003	2004	Mean
1. Huarng et al. [36] (Using NASDAQ)	NA	158.7	136.49	95.15	65.51	73.57	105.88
2. Huarng et al. [36] (Using Dow Jones)	NA	165.8	138.25	93.73	72.95	73.49	108.84
3. Huarng et al. [36] (Using M_{1b})	NA	160.19	133.26	97.1	75.23	82.01	111.36
4. Huarng et al. [36] (Using NASDAQ and M_{1b})	NA	157.64	131.98	93.48	65.51	73.49	104.42
5. Huarng et al. [36] (Using Dow Jones and M_{1b})	NA	155.51	128.44	97.15	70.76	73.48	105.07
6. Huarng et al. [36] (Using NASDAQ, Dow Jones and M_{1b})	NA	154.42	124.02	95.73	70.76	72.35	103.46
7. Chen et al. [21, 31, 32]	120	176	148	101	74	84	117.4
8. U_R model [31, 32]	164	420	1070	116	329	146	374.2
9. U_NN model [31, 32]	107	309	259	78	57	60	145.0
10. U_NN_FTS model [27, 31, 32]	109	255	130	84	56	116	125.0
11. U_NN_FTS_S model [27, 31, 32]	109	152	130	84	56	116	107.8
12. B_R model [31, 32]	103	154	120	77	54	85	98.8
13. B_NN model [31, 32]	112	274	131	69	52	61	116.4
14. B_NN_FTS model [31, 32]	108	259	133	85	58	67	118.3
15. B_NN_FTS_S model [31, 32]	112	131	130	80	58	67	96.4
16. Chen et al. [47] (Using Dow Jones)	115.47	127.51	121.98	74.65	66.02	58.89	94.09
17. Chen et al. [47] (Using NASDAQ)	119.32	129.87	123.12	71.01	65.14	61.94	95.07
18. Chen et al. [47] (Using M_{1b})	120.01	129.87	117.61	85.85	63.1	67.29	97.29
19. Chen et al.[47] (Using NASDAQ and Dow Jones)	*116.64*	*123.62*	*123.85*	*71.98*	*58.06*	*57.73*	*91.98*
20. Chen et al. [47] (Using Dow Jones and M_{1b})	116.59	127.71	115.33	77.96	60.32	65.86	93.96
21. Chen et al. [47] (Using NASDAQ and M_{1b})	114.87	128.37	123.15	74.05	67.83	65.09	95.56
22. Chen et al. [47] (Using NASDAQ, Dow Jones and M_{1b})	112.47	131.04	117.86	77.38	60.65	65.09	94.08
23. Karnik-Mendel [44] induced stock prediction	116.60	128.46	120.62	78.60	66.80	68.48	96.59

(continued)

Table 2.5 (continued)

Methods	Years						
	1999	2000	2001	2002	2003	2004	Mean
24. Chen et al. [50] (Using NASDAQ, Dow Jones and M_{1b})	101.47	122.88	114.47	67.17	52.49	52.84	85.22
25. Chen et al. [59]	87.67	125.34	114.57	76.86	54.29	58.17	86.14
26. Cai et al. [60]	102.22	131.53	112.59	60.33	51.54	50.33	84.75
27. Mu-Yen Chen [61]	NA	108	88	60	42	NA	74.5
28. Proposed method 1	114.20	127.12	110.50	70.56	52.10	48.40	87.18
29. Proposed method 2	101.84	125.87	111.60	71.66	52.77	50.166	85.654
30. Proposed method 3	92.665	108.18	105.51	66.99	47.11	43.83	77.382
31. Proposed method 4	94.610	113.04	110.81	66.04	48.77	46.179	79.910
32. Proposed method 5	89.021	99.765	101.71	58.32	37.12	36.600	70.424

triangular MFs are illustrated in Fig. 2.8b. Both the realizations confirm that adaptation has merit in the context of prediction, irrespective of the choice of MFs.

2.4.2.4 Varying *d*

We also study the effect of variation of '*d*' [62, 63] on the results of forecasting using proposed method 5. Here, for each integer value of d in [1, 4], we obtain the plots of actual and forecasted close price as indicated in Fig. 2.5. The fuzzy logical implication rules and frequency of occurrences from CSVS to MFVS are determined using Fig. 2.9 following similar approach as done for $d = 1$. These rules and frequency of occurrences are given in Tables 2.7 and 2.14 (SEE APPENDIX). It is apparent from Fig. 2.9 that forecasted price with delay $d = 1$ yields an RMSE of 36.6006, which is found to be smallest among the considered RMSEs for d = 1, 2, 3 and 4. This indicates that setting $d = 1$ returns the best possible prediction, which also has logical justification in the sense that the predicted close price intricately depends on close price of yesterday, rather than that of day before yesterday or its preceding days.

2.5 Performance Analysis

This section attempts to compare the relative performance of the proposed five techniques with 27 other techniques [21, 27, 31, 32, 36, 47] using RMSE as the metric for comparison. Table 2.5 provides the results of comparison for the period 1999–2004 with mean and standard deviation of all the RMSEs obtained for the above period. It is apparent from Table 2.5 that the entries in the last row are

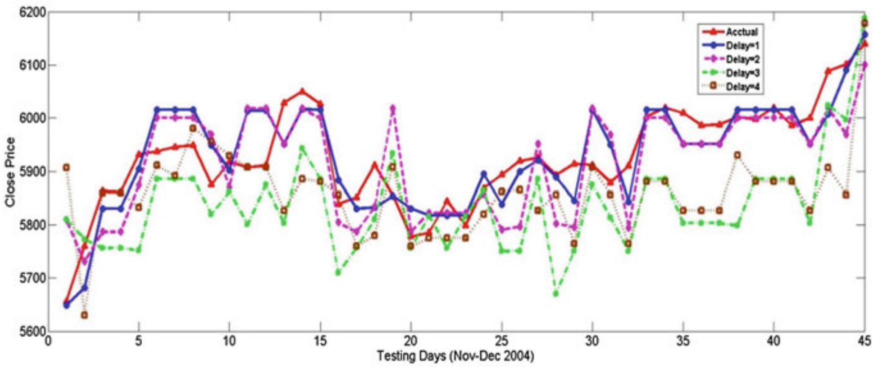


Fig. 2.9 Forecasted TAIEX of the months November and December 2004 using Proposed method 5 for different values of d. The respective RMSEs are RMSE = 36.6006 for d = 1, RMSE = 72.8012 for d = 2, RMSE = 122.4201 for d = 3, RMSE = 140.2005 for d = 4

smaller than the entries above. This indicates that that RMSE for each column on the last row of Table 2.5 being the smallest, the proposed method 5 seems to outperform the other techniques (calculated with respect of mean of 6 years RMSE) by at least 23%, encountered in method-19 in Table 2.5.

We here use paired t-test [64] to examine the statistical confidence on the results of prediction by different algorithms using RMSE as the metric. Let, H_o be the null hypothesis to compare two algorithms' performance, where one is the reference algorithm, while the other is any one of the existing algorithms. Here, we consider the proposed algorithm as the reference algorithm. Thus, $H_o =$ Performance of algorithm A and reference algorithm R are comparable.

Let A be the algorithm by Chen et al. [47]. To statistically validate the Hypothesis H_o, we evaluate t-measure, given by

$$t = \frac{(m_A - m_R)}{\sqrt{{s_A}^2 + {s_R}^2}}, \tag{2.15}$$

where m_A and m_R are the mean values of the distributions of RMSE obtained by algorithms A and R respectively with equal sample size in the two distributions, and s_A and s_R are the standard deviations of the respective samples obtained by algorithms A and R.

After evaluation of statistic t, we consult the t-Table (Table 2.6) with degrees of freedom $KI =$ sample size of any one population minus $1 = n - 1$, say. Let the value obtained from the t-Table for given confidence level α and KI be z. Now, if $z < t$, the calculated value by formula (2.15), then the H_o is wrong, and its contradiction that the proposed algorithm is better than A with respect to RMSE is true. We now repeat the above steps for different comparative algorithms A and found that $z < t$ always holds, thereby indicating that the proposed algorithm outperforms all other existing algorithms.

Table 2.6 Results of statistical significance with the proposed method 1–5 as the reference, one at a time (t-table)

	Statistical significance				
Existing methods		Reference methods			
	Proposed method 1	Proposed method 2	Proposed method 3	Proposed method 4	Proposed method 5
1	−	−	+	+	+
2	+	+	+	+	+
3	+	+	+	+	+
4	+	+	+	+	+
5	+	+	+	+	+
6	−	+	+	+	−
7	−	+	+	+	+

(continued)

Table 2.6 (continued)

	Statistical significance				
Existing methods		Reference methods			
	Proposed method 1	Proposed method 2	Proposed method 3	Proposed method 4	Proposed method 5
8	+	+	−	+	−
9	+	−	−	−	−
10	−	−	+	−	−
11	−	+	+	+	+
12	+	+	+	+	+
13	+	−	−	−	−
14	−	−	−	−	−
15	+	+	+	+	+
16	+	+	+	+	+
17	+	+	+	−	+
18	+	+	+	+	+
19	+	−	−	+	+
20	+	+	−	+	+
21	+	+	−	+	+
22	−	−	+	−	+
23	+	+	+	+	+
24	−	+	+	+	−
25	−	+	+	+	+
26	−	+	−	+	+
27	+	+	+	−	+

Table 2.6 is designed to report the results of statistical test considering proposed method 1–5 as the reference. The degree of freedom is here set to 5 as the prediction data set used involves six years' RMSE data. The plus (minus) sign in Table 2.6 represents that the difference of means of an individual method with the proposed method as reference is significant (not significant). The degree of significance here is studied at 0.05 level, representing 95% confidence level.

2.6 Conclusion

This chapter introduced a novel approach to stock index time-series prediction using IT2Fs. Such representation helps overcoming the possible hindrances in stock index prediction as introduced in the introduction. Both triangular and Gaussian MFs along with provision of their adaptation have been introduced to examine their relative performance in prediction. The strategy used to consider secondary to main factor variation has considerably improved the relative performance of the stock index time-series prediction. A thorough analysis of results using RMSE as the metric indicates that the proposed methods outperform the existing techniques on stock index prediction by a considerable margin ($\geq 23\%$). Out of the five proposed methods, the method employing triangular MF with provision for its adaptation yields the best performance following the prediction of TAIEX stock data for the period of 1999–2004 with DOWJONES and NASDAQ together as the composite secondary index. A statistical analysis undertaken with paired t-test confirms that each of the proposed algorithms outperforms most of the existing algorithms with root mean square error as the key metric at 95% confidence level. With an additional storage of fuzzy logical implication rules and frequency of occurrences from CSVS to MFVS for $d = 1, 2, \ldots, k$, we would be able to predict the close price on the next day, next to next day and the like from today's close price. Further extension of the proposed technique can be accomplished by using General Type-2 fuzzy sets, which is expected to improve performance at the expense of additional complexity.

2.7 Exercises

1. Graphically plot the interval type-2 fuzzy set constructed from type-2 membership functions in Fig. 2.10.

[**Hints:** The UMF and LMF constructed from the given type-1 MFs are given in Fig. 2.11.]

2. Construct the rules from a partitioned time-series, indicated in Fig. 2.12.

[**Hints:** The rules following the occurrence of the data point in the partition are: $P_1 \rightarrow P_2$, $P_2 \rightarrow P_4$, $P_4 \rightarrow P_2$, $P_2 \rightarrow P_1$.]

3. Let there be three partitions P_1, P_2, P_3 of a stock data of a stock data, the corresponding fuzzy sets are A_1, A_2, and A_3. Suppose we have the rules: $A_1 \rightarrow A_2$ and $A_1 \rightarrow A_3$ as indicated below, Determine the stock price of tomorrow if the stock price of today, as indicated falls in partition P_1 (i.e. fuzzy set of P_1). Presume that, $\sum_x x\_movement$ of the inferred membership function is 100 with

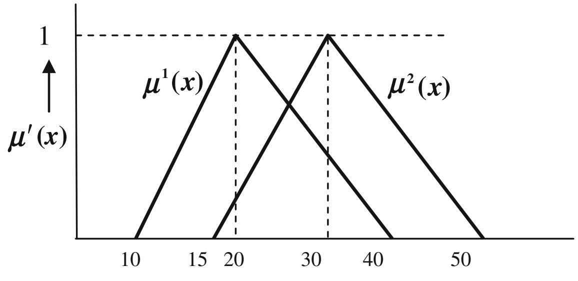


Fig. 2.10 Figure for Problem 1

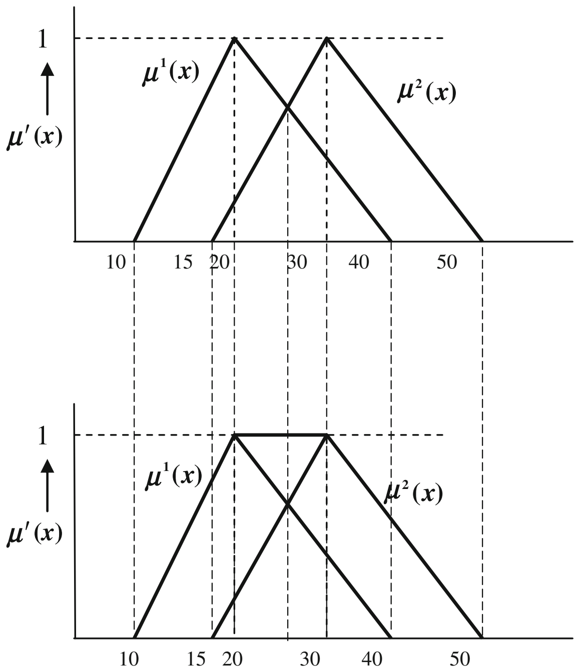


Fig. 2.11 Solution for Problem 1

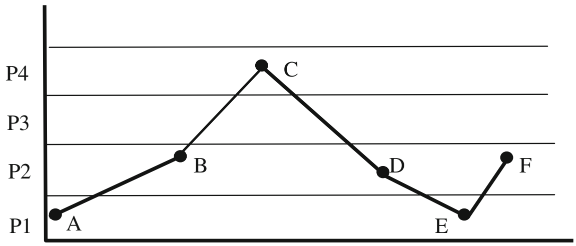


Fig. 2.12 Figure for Problem 2

and area under the inferred membership = 12 unit [Ans: 100/12 = 8.33] (Fig. 2.13).

4. Let the inferred membership function be a sine function for x = 0 to π. Find the centroid. Refer Figs. 2.14 and 2.15.

$$\text{[}\textbf{Hints}\text{: Centroid} = \frac{\int_0^{\pi} x \sin x dx}{\int_0^{\pi} \sin x dx} = \frac{[x(-\cos x) + \sin x]_0^{\pi}}{[-\cos \pi]_0^{\pi}} = \frac{\pi}{1+1} = \pi/2\text{]}$$

5. Let the partitions be P_1, P_2 and, also the IT2FS used for three partitions be $\tilde{A}_1, \tilde{A}_2$ and $\tilde{A}_3$ respectively. Given the IT2FS for the stock data and the rules $A_1 \rightarrow A_2$ and $A_1 \rightarrow A_3$. If today's stock price falls in $\tilde{A}_3$, then will you be able to generate the fuzzy inference for tomorrow.

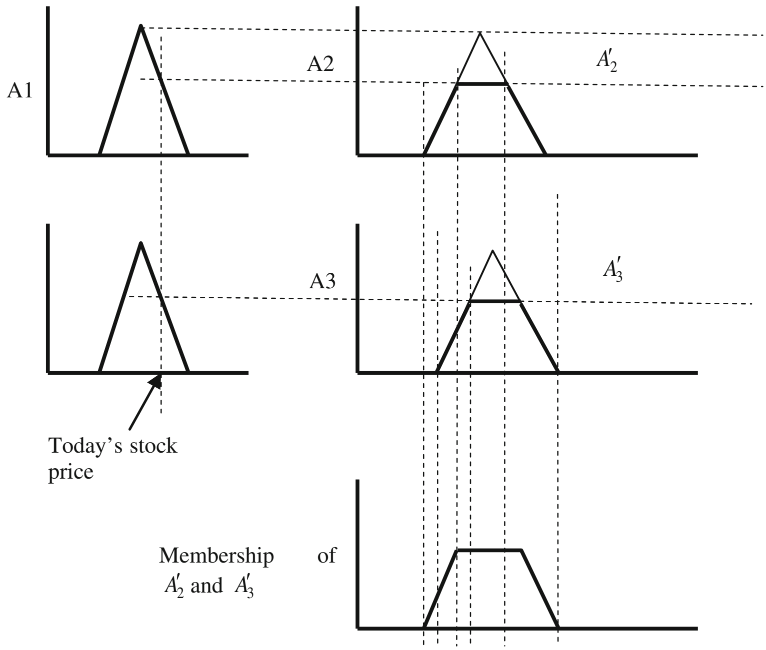
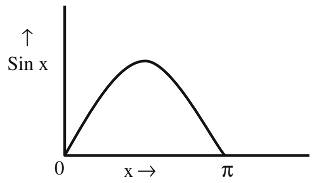


Fig. 2.13 Figure for Problem 3

Fig. 2.14 Figure for Problem 4



[**Hints:** The inference generation is examined below. Add figure given in Fig. 2.16]

6. In stock index prediction, we use the secondary factor, here DOW JONES stock index data for the main factor TAIEX time-series as indicated in Fig. 2.17. On the day (t − 1), it is observed that the secondary index lies in partition B_5, while the main factor time-series falls in A_3. Given the rules under group B_5 (Fig. 2.18):

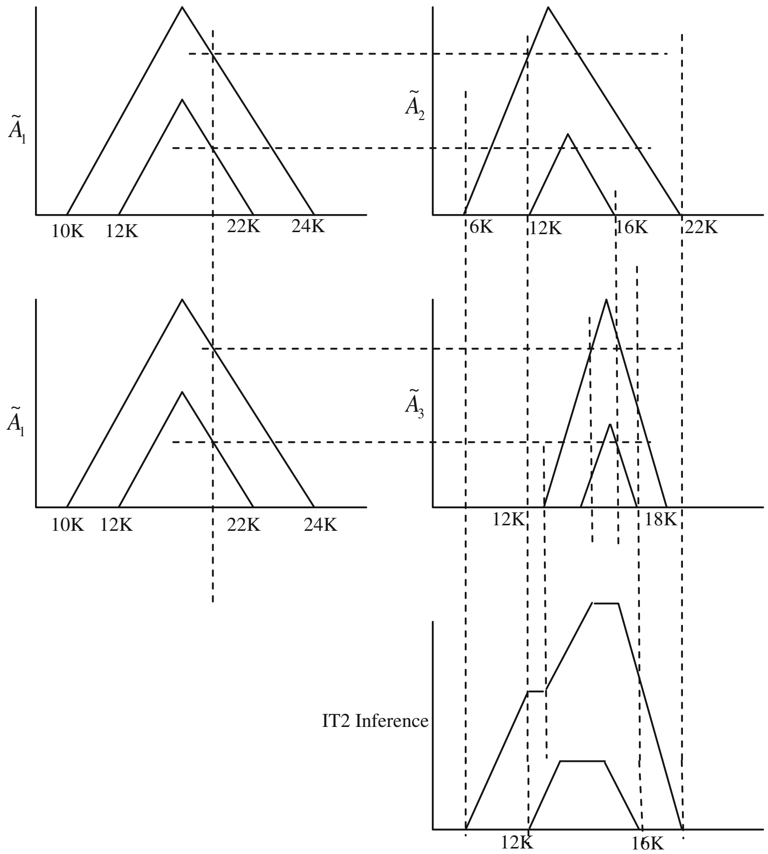


Fig. 2.15 IT2 Inference generation for Problem 4

$$
\begin{aligned}
&A_1 \rightarrow \\
&A_2 \rightarrow \\
&A_3 \rightarrow A_4, A_6 \\
&A_4 \rightarrow \\
&\ldots \\
&A_{5\rightarrow}
\end{aligned}
$$

Given the MFs of A_3, A_4, A_6 as follows and today's price as 1100. Determine the fuzzy inference (Fig. 2.19).

7. A close price time-series is partitioned into 4 partitions: P_1, P_2, P_3 and P_4. The close price falling in P_i would have a membership 1 in A_i fuzzy sets and membership 0.5 in A_{i-1} and A_{i+1} and zero elsewhere. If the range of P_1, P_2, P_3 and P_4 are [0, 1 K), [1 K, 2 K), [2 K, 3 K) and [3 K, 4 K] respectively, construct A_1, A_2, A_3 and A_4 (Fig. 2.20).

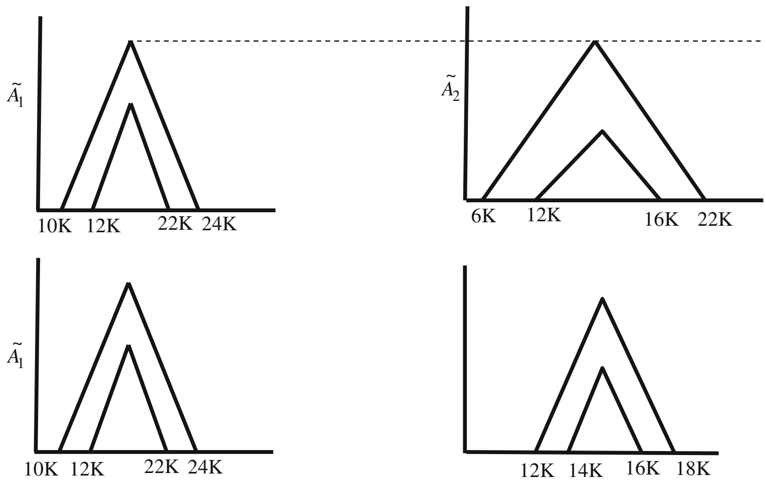


Fig. 2.16 Figure for Problem 5

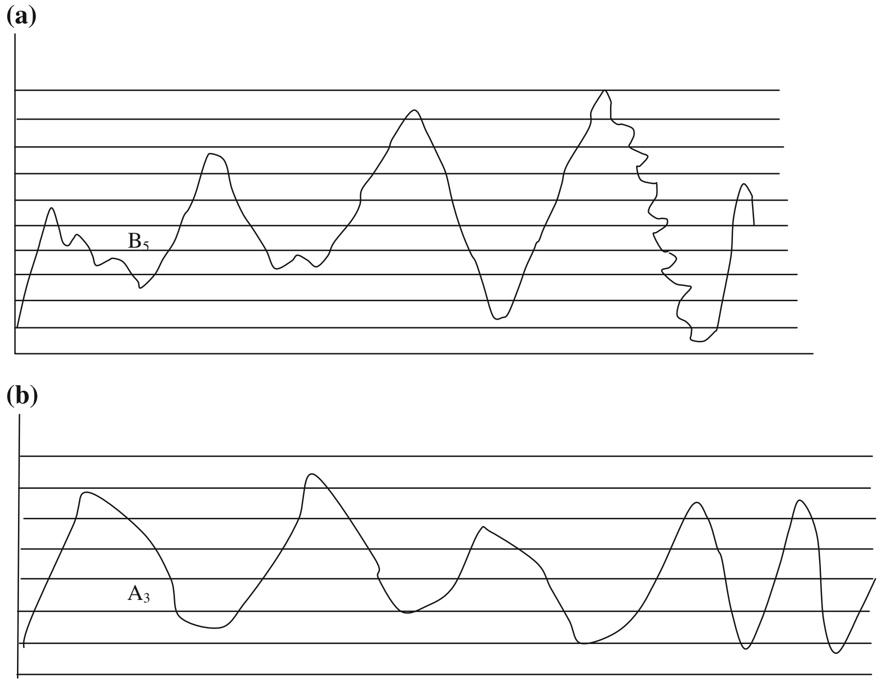


Fig. 2.17 Figure for Problem 6

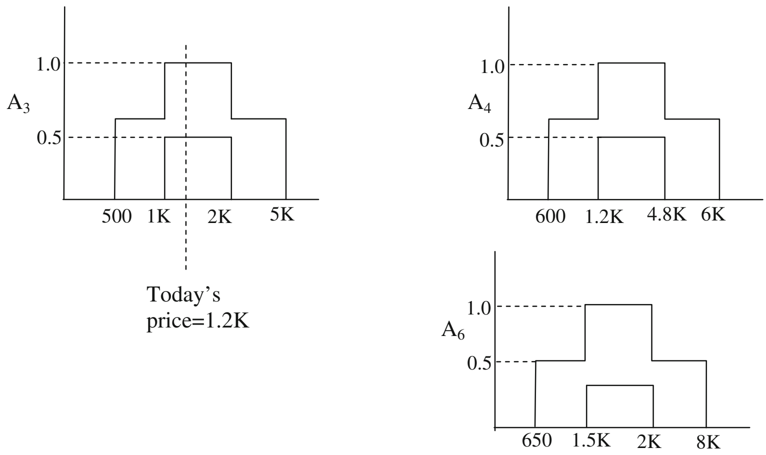


Fig. 2.18 Figure for Problem 6

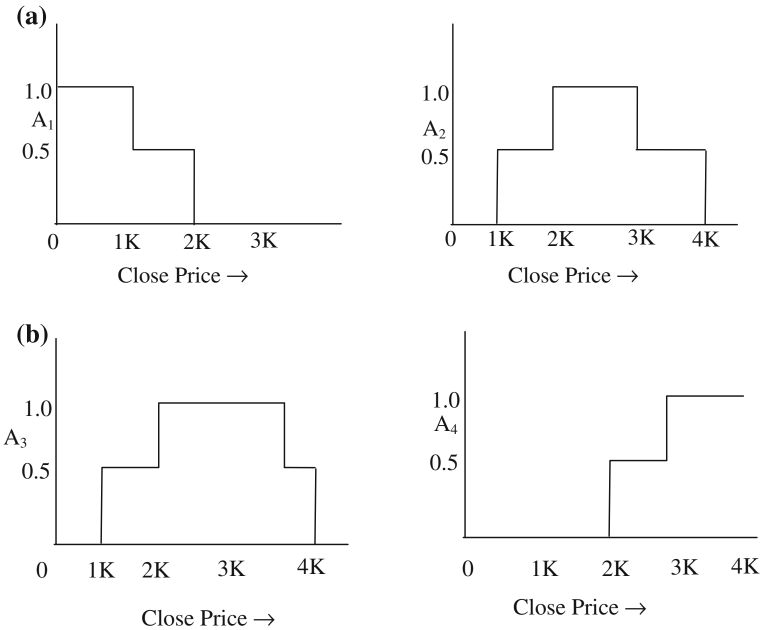


Fig. 2.19 Figure for Problem 7

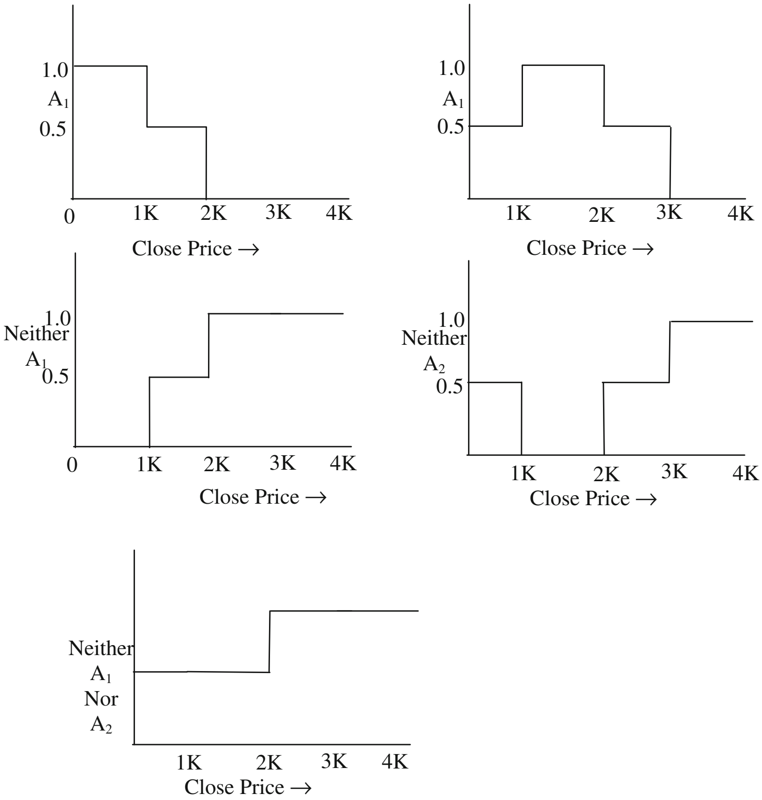


Fig. 2.20 Computation of Neither A1 Nor A2

[**Hints**: membership functions are given below.]

8. In question 7 suppose we need to construct the membership function of (i) neither A_1 nor A_2, (ii) either A_1 or A_3, (iii) neither A_1 nor A_2 and A_3.

[**Hints**: We show the solution for part (i). The rest can be obtained similarly.]

Appendix 2.1

See Tables 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12 and 2.13.

Table 2.7 Fuzzy logical implication rules for year 2004

Group	Fuzzy logical implication
B_3	$A_3 \to A_2$
B_4	$A_2 \to A_1, A_2$; $A_3 \to A_2$; $A_5 \to A_4$
B_5	$A_1 \to A_1$; $A_4 \to A_4, A_5$; $A_2 \to A_2$; $A_6 \to A_6, A_7$; $A_3 \to A_3$;$A_8 \to A_8$
B_6	$A_1 \to A_1, A_1$; $A_7 \to A_6$; $A_2 \to A_1, A_1, A_2, A_2$;$A_8 \to A_8$; $A_3 \to A_2, A_2, A_3, A_3$; $A_9 \to A_8$; $A_4 \to A_3$; $A_{10} \to A_{10}$; $A_5 \to A_5, A_5, A_5$;$A_{11} \to A_{10}$;$A_6 \to A_6$
B_7	$A_1 \to A_1, A_1, A_1$; $A_7 \to A_7, A_7, A_7, A_7, A_7, A_7$; $A_2 \to A_2, A_2, A_2, A_2, A_2, A_2, A_2$; $A_8 \to A_8, A_8, A_8, A_8, A_8, A_8, A_8, A_8, A_8$; $A_3 \to A_2, A_2, A_2, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3$; $A_4 \to A_4, A_4, A_4, A_4$;$A_9 \to A_8, A_8, A_8, A_9$; $A_5 \to A_4, A_5, A_5, A_5, A_5, A_5$; $A_{10} \to A_{10}, A_{10}, A_{10}, A_{10}, A_{10}$;
B_8	$A_1 \to A_1, A_1, A_1, A_1, A_1$; $A_6 \to A_6, A_6, A_6, A_6$; $A_2 \to A_2, A_2, A_2, A_2, A_2, A_2, A_3$; $A_7 \to A_7, A_7, A_7, A_8$; $A_3 \to A_3, A_3, A_3, A_3, A_3, A_4$; $A_8 \to A_8, A_8, A_8, A_8, A_8, A_8, A_8, A_9, A_9, A_9, A_9$; $A_4 \to A_4, A_4, A_4, A_4, A_4$; $A_9 \to A_9, A_9, A_9, A_9, A_9$; $A_5 \to A_5, A_5, A_5, A_5, A_5, A_5, A_5, A_5, A_5$;
B_9	$A_1 \to A_1, A_1, A_1, A_1, A_2, A_2, A_2$; $A_7 \to A_7, A_7, A_7$;$A_2 \to A_2, A_2, A_3, A_3$; $A_8 \to A_8, A_8, A_9$;$A_3 \to A_3, A_3, A_3, A_4$;$A_9 \to A_9, A_{10}$; $A_4 \to A_4, A_4, A_5$;$A_{10} \to A_{10}, A_{10}$;$A_5 \to A_5, A_5, A_6$
B_{10}	$A_2 \to A_2, A_2, A_3, A_3$;$A_7 \to A_7, A_7$;$A_3 \to A_3, A_3$; $A_8 \to A_9, A_9$;$A_4 \to A_4, A_5, A_5$
B_{11}	$A_2 \to A_2$;$A_6 \to A_7, A_7$;$A_3 \to A_3$
B_{12}	$A_2 \to A_3, A_4$

Table 2.8 Fuzzy logical implication rules considering d = 2

Group	Fuzzy logical relationship
B_1	$A_3 \to A_2$; $A_6 \to A_4$, A_5; $A_8 \to A_5$;
B_2	$A_3 \to A_2, A_3$; $A_5 \to A_5$;
B_3	$A_3 \to A_4$;
B_4	$A_1 \to A_1, A_3$;$A_3 \to A_2, A_3$; $A_4 \to A_1$; $A_7 \to A_5, A_7$; $A_6 \to A_7$;
B_5	$A_2 \to A_1, A_2, A_2, A_2, A_4$;$A_3 \to A_3, A_4 A_4 \to A_3, A_3, A_3, A_4$; $A_5 \to A_4, A_5$; $A_7 \to A_6$;

(continued)

Table 2.8 (continued)

Group	Fuzzy logical relationship
B_6	$A_1 \rightarrow A_1, A_1; A_2 \rightarrow A_2, A_2; A_3 \rightarrow A_2, A_2, A_2, A_3, A_3, A_3, A_3, A_3, A_3;$ $A_4 \rightarrow A_2, A_3, A_3, A_3; A_5 \rightarrow A_5, A_6; A_6 \rightarrow A_5, A_6; A_7 \rightarrow A_7;$
B_7	$A_1 \rightarrow A_1, A_1, A_1, A_1, A_1, A_1, A_1, A_1, A_1, A_1;$ $A_2 \rightarrow A_1, A_2, A_3, A_3;$ $A_3 \rightarrow A_2, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3;$ $A_4 \rightarrow A_3, A_3, A_4, A_5; A_5 \rightarrow A_4, A_4, A_5, A_5, A_5;$ $A_6 \rightarrow A_6, A_6, A_7; A_7 \rightarrow A_7, A_7, A_7, A_8, A_8; A_6 \rightarrow A_6, A_6, A_7;$ $A_7 \rightarrow A_7, A_7, A_7, A_8, A_8; A_8 \rightarrow A_7, A_8, A_8, A_8, A_8, A_8;\ A_9 \rightarrow A_8$
B_8	$A_1 \rightarrow A_1, A_1, A_1, A_1, A_1; A_2 \rightarrow A_2, A_2, A_2, A_3, A_3; A_8 \rightarrow A_8, A_8$ $A_3 \rightarrow A_2, A_3, A_3, A_4; A_4 \rightarrow A_3, A_3, A_5, A_5; A_9 \rightarrow A_8, A_9, A_9, A_9$ $A_5 \rightarrow A_4, A_5, A_5, A_5, A_5, A_5, A_5, A_5, A_5, A_5, A_6, A_6$ $A_6 \rightarrow A_5, A_7; A_7 \rightarrow A_7, A_7, A_7, A_7, A_7, A_7;$
B_9	$A_1 \rightarrow A_1, A_1, A_1, A_2; A_2 \rightarrow A_2; A_3 \rightarrow A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_4, A_4$ $A_4 \rightarrow A_3, A_3, A_4, A_4, A_4, A_5; A_5 \rightarrow A_5, A_5, A_6, A_6;$ $A_6 \rightarrow A_6, A_7; A_7 \rightarrow A_7, A_7, A_7, A_8, A_8, A_8; A_8 \rightarrow A_8$
B_{10}	$A_2 \rightarrow A_3; A_3 \rightarrow A_3, A_3, A_4, A_4, A_4; A_4 \rightarrow A_4, A_4; A_6 \rightarrow A_6; A_7 \rightarrow A_7, A_8; A_8 \rightarrow A_8, A_9;$
B_{11}	$A_2 \rightarrow A_2, A_3; A_4 \rightarrow A_3, A_4; A_6 \rightarrow A_6; A_8 \rightarrow A_6$
B_{12}	$A_1 \rightarrow A_2; A_3 \rightarrow A_3, A_4, A_4; A_4 \rightarrow A_4$
B_{13}	$A_5 \rightarrow A_6; A_6 \rightarrow A_7;$
B_{14}	$A_2 \rightarrow A_3;$

Table 2.9 Fuzzy logical implication rules considering d = 3

Group	Fuzzy logical relationship
B_1	$A_1 \rightarrow A_3; A_3 \rightarrow A_2, A_3;$ $A_5 \rightarrow A_3, A_5; A_6 \rightarrow A_5, A_5;$
B_2	$A_3 \rightarrow A_4;$
B_3	$A_2 \rightarrow A_4; A_3 \rightarrow A_2; A_4 \rightarrow A_3, A_3;$ $A_5 \rightarrow A_4; A_7 \rightarrow A_4, A_8$
B_4	$A_1 \rightarrow A_1; A_7 \rightarrow A_8; A_2 \rightarrow A_1, A_1, A_2, A_2; A_8 \rightarrow A_5, A_7;$ $A_3 \rightarrow A_3, A_3; A_4 \rightarrow A_4; A_5 \rightarrow A_6;$
B_5	$A_2 \rightarrow A_1, A_2, A_3; A_3 \rightarrow A_2, A_3, A_3;$ $A_4 \rightarrow A_3, A_3, A_3; A_5 \rightarrow A_6; A_7 \rightarrow A_5; A_8 \rightarrow A_7;$
B_6	$A_1 \rightarrow A_1, A_1, A_1, A_1; A_2 \rightarrow A_2, A_3, A_3;$ $A_3 \rightarrow A_3, A_3, A_3, A_3, A_4, A_4;$ $A_4 \rightarrow A_2, A_2; A_6 \rightarrow A_5; A_7 \rightarrow A_7; A_8 \rightarrow A_6, A_7, A_8;$
B_7	$A_1 \rightarrow A_1, A_1, A_1, A_1, A_1, A_1; A_2 \rightarrow A_3;$ $A_3 \rightarrow A_2, A_2, A_2, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_3, A_4, A_4, A_4, A_4;$ $A_4 \rightarrow A_2, A_3, A_4; A_5 \rightarrow A_3, A_5, A_6, A_6;$ $A_6 \rightarrow A_5, A_5, A_7;$ $A_7 \rightarrow A_8, A_8, A_8; A_8 \rightarrow A_7, A_7, A_8, A_8, A_8; A_9 \rightarrow A_8, A_8, A_8, A_9;$
B_8	$A_1 \rightarrow A_1, A_1, A_1, A_1, A_1, A_1, A_1, A_2;$ $A_2 \rightarrow A_3;$ $A_3 \rightarrow A_2, A_2, A_3, A_3, A_3, A_3, A_4;$

(continued)

Table 2.9 (continued)

Group	Fuzzy logical relationship
	$A_4 \to A_3, A_4$; $A_5 \to A_4, A_4, A_5, A_5, A_5, A_5, A_5, A_5, A_5, A_6, A_6$; $A_6 \to A_6, A_7$;$A_7 \to A_7, A_7, A_7, A_7, A_7, A_8$; $A_8 \to A_8, A_8$;
B_9	$A_1 \to A_1, A_1$;$A_2 \to A_2, A_3, A_4$; $A_3 \to A_3, A_3, A_3, A_3, A_4, A_4, A_4$; $A_4 \to A_1, A_3, A_3, A_4, A_4, A_5, A_5$; $A_5 \to A_5, A_5, A_5, A_6, A_6$; $A_6 \to A_6, A_7$;$A_7 \to A_7, A_7, A_7, A_7, A_8$;$A_8 \to A_8$;
B_{10}	$A_2 \to A_3, A_3$;$A_3 \to A_2, A_3, A_3$;$A_4 \to A_3, A_4$; $A_7 \to A_6, A_7, A_7, A_9$;$A_9 \to A_9, A_9$;
B_{11}	$A_4 \to A_4$;$A_6 \to A_6, A_6$;
B_{12}	$A_1 \to A_2$;$A_2 \to A_2$;$A_3 \to A_3, A_3$; $A_4 \to A_3, A_4, A_5$;$A_5 \to A_7$;$A_8 \to A_9$;
B_{13}	$A_2 \to A_2$;$A_4 \to A_3$;$A_6 \to A_7$;
B_{14}	$A_3 \to A_4, A_4$;$A_6 \to A_7$;

Table 2.10 Fuzzy logical implication rules considering d = 4

Group	Fuzzy logical relationship
B_1	$A_1 \to A_4$; $A_3 \to A_2, A_3, A_4$; $A_5 \to A_3, A_4, A_6, A_6$; $A_6 \to A_3, A_5$
B_2	$A_1 \to A_1$; $A_2 \to A_2, A_2$; $A_4 \to A_4, A_4$; $A_7 \to A_5$
B_3	$A_2 \to A_1, A_1, A_4$; $A_3 \to A_3$; $A_4 \to A_3$; $A_7 \to A_6, A_8$; $A_8 \to A_8$
B_4	$A_2 \to, A_1$; $A_8 \to A_5$; $A_3 \to A_2, A_3, A_3$; $A_5 \to A_7$;
B_5	$A_1 \to A_1, A_1, A_1$; $A_3 \to A_3, A_3, A_4, A_4$; $A_4 \to A_2, A_3, A_4$; $A_6 \to A_6$; $A_7 \to A_4$; $A_8 \to A_5$; $A_9 \to A_7$;
B_6	$A_1 \to A_1$; $A_2 \to A_1, A_2, A_3, A_3$; $A_3 \to A_3, A_3, A_3, A_4$; $A_4 \to A_3$; $A_5 \to A_6, A_6$; $A_7 \to A_5$; $A_8 \to A_6, A_7, A_7, A_8$; $A_9 \to A_8$;
B_7	$A_1 \to A_1, A_1, A_1, A_1, A_1, A_1, A_1, A_1, A_2, A_2$; $A_2 \to A_3, A_3, A_4$; $A_3 \to A_2, A_2, A_2, A_2, A_3, A_3, A_3, A_3, A_4, A_4, A_4$; $A_4 \to A_3, A_3$; $A_5 \to A_5$; $A_7 \to A_5, A_8, A_8, A_8$; $A_8 \to A_7, A_7, A_8$;
B_8	$A_1 \to A_1, A_1, A_2$; $A_2 \to A_3, A_3$; $A_3 \to A_2, A_3, A_3, A_3, A_3, A_4, A_4$; $A_4 \to A_3, A_3, A_3, A_4, A_5$; $A_5 \to A_5, A_5, A_5, A_5, A_6$; $A_6 \to A_5, A_6, A_7, A_7$; $A_7 \to A_7, A_7, A_7, A_7, A_8, A_8, A_9$; $A_8 \to A_8, A_8$;
B_9	$A_1 \to A_1, A_1$; $A_2 \to A_2, A_3, A_4$; $A_3 \to A_3, A_3, A_3, A_3, A_4, A_4, A_4$; $A_4 \to A_1, A_3, A_3, A_4, A_4, A_5, A_5$; $A_5 \to A_5, A_5, A_5, A_6, A_6$; $A_6 \to A_6, A_7$; $A_7 \to A_7, A_7, A_7, A_7, A_8$; $A_8 \to A_8$;
B_{10}	$A_3 \to A_2, A_3, A_3, A_3$; $A_4 \to A_2, A_3, A_4$; $A_5 \to A_5$; $A_6 \to A_7$; $A_7 \to A_7, A_7, A_7, A_7, A_8$; $A_8 \to A_5, A_8$;
B_{11}	$A_1 \to A_2$; $A_2 \to A_3$; $A_3 \to A_3$; $A_4 \to A_3, A_4, A_4, A_5$; $A_6 \to A_6, A_7$; $A_8 \to A_8$; $A_9 \to A_9$;
B_{12}	$A_2 \to A_3$; $A_3 \to A_3$; $A_4 \to A_5$; $A_6 \to A_7$; $A_8 \to A_9, A_9$; $A_9 \to A_9$;
B_{13}	$A_2 \to A_2, A_3$; $A_4 \to A_2$; $A_6 \to A_7$;
B_{14}	$A_3 \to A_4$; $A_4 \to A_4$; $A_7 \to A_7$

Table 2.11 Frequency of occurrence of main factor variation in each group of secondary factor variation considering d = 1

		Main factor variation													
		B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
Secondary factor variation	B'_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_4	0	0	0	0	0	2	0	0	0	0	0	0	0	0
	B'_5	0	0	0	0	1	1	2	1	0	0	0	0	0	0
	B'_6	0	0	0	1	2	5	9	10	0	0	0	0	0	0
	B'_7	0	0	0	2	2	7	19	13	11	1	1	0	0	0
	B'_8	0	0	1	0	4	3	24	23	10	8	1	0	0	0
	B'_9	0	0	0	1	0	1	9	10	8	1	2	0	0	0
	B'_{10}	0	0	0	0	0	1	1	4	1	1	0	1	0	0
	B'_{11}	0	0	0	0	0	0	0	0	1	0	0	1	0	0
	B'_{12}	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	B'_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.12 Frequency of occurrence of main factor variation in each group of secondary factor variation considering d = 2

		Main factor variation													
		B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
Secondary factor variation	B'_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_6	2	1	0	3	3	4	12	2	4	1	0	1	0	1
	B'_7	2	0	0	3	8	11	15	18	9	2	3	0	1	0
	B'_8	0	2	1	1	3	10	22	17	12	7	2	4	0	0
	B'_9	0	0	0	1	0	1	3	7	7	3	2	0	0	0
	B'_{10}	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	B'_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.13 Frequency of occurrence of main factor variation in each group of secondary factor variation considering d = 3

		Main factor variation													
		B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
Secondary factor variation	B'_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_5	1	0	0	0	0	1	2	2	0	1	1	0	0	0
	B'_6	2	0	2	3	4	5	7	4	6	0	0	0	1	1
	B'_7	3	0	1	4	4	4	19	7	6	7	0	1	0	1
	B'_8	1	1	4	3	3	7	12	15	14	3	4	4	1	0
	B'_9	0	0	0	1	1	3	6	11	2	2	0	3	0	1
	B'_{10}	0	0	0	1	0	1	0	1	3	0	1	1	0	0
	B'_{11}	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	B'_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2.14 Frequency of occurrence of main factor variation in each group of secondary factor variation considering d = 4

		Main factor variation													
		B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}
Secondary factor variation	B'_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_6	2	2	3	2	4	7	7	2	1	1	1	0	1	1
	B'_7	4	1	2	2	5	5	11	14	15	5	3	4	0	1
	B'_8	2	3	3	1	5	8	12	14	13	8	7	3	1	1
	B'_9	2	0	0	1	1	0	4	5	6	1	0	1	1	0
	B'_{10}	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	B'_{11}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B'_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Appendix 2.2: Source Codes of the Programs

% MATLAB Source Code of the Main Program and Other Functions for Time
% Series Prediction by IT2FS REasoning

% Developed by Monalisa Pal

% Under the guidance of Amit Konar and Diptendu Bhattacharya

```
% Main Program
%___________________________________________________________________
clear all;
clc;
str=input('Data file having Main Factor:');
load(str);
closeMF=close;
n=input('Number of secondary factors:');
n=str2double(n);
SF=zeros(length(closeMF),n);
for i=1:n
  strSF=input(['Data file having ',num2str(i),'-th   Secondary Factor:']);
  load(strSF);
  SF(:,i)=close;
end
clear close;
tic;
%% Training
[A,B,VarMF,Au,Al]=partitioning(closeMF);
plotpartitions(closeMF(s:e1),A(s:e1),Al,Au);
FLRG=tableVI(B(2:e1),A(1:e1-1),A(2:e1));
VarSF=overallvar(VarMF,SF,e1,n);
BSF=fuzzyvarSF(VarSF);
FVG=tableIX(B(2:e1),BSF(2:e1));
WBS=tableX(FVG);
%% Validation (Differential Evolution)
sd=extractSD(A(s:e1),closeMF(s:e1));
[UMF,LMF,typeMF,rmse]=MFusingDE(A(s:e1),closeMF(s:e1),
Al:Au-1,sd,closeMF(e1+1:e2),A(e1:e2),FLRG,WBS,BSF(e1:e2));
plotDE(rmse);
plotFOU(UMF,LMF,Al:Au-1);
%% Inference
forecasted=predict(closeMF(e2+1:f),A(e2:f),Al,Au,UMF,LMF,typeMF,
FLRG,WBS,BSF(e2:f));
[CFE,ME,MSE,RMSE,SD,MAD,MAPE]=
errormetrics(forecasted,closeMF(e2+1:f));
```

```
disp('CFE');disp(CFE);
disp('ME');disp(ME);
disp('MSE');disp(MSE);
disp('RMSE');disp(RMSE);
disp('SD');disp(SD);
disp('MAD');disp(MAD);
disp('MAPE');disp(MAPE);
plotforecasted(forecasted,closeMF(e2+1:f),RMSE);
comp_time=toc;
disp('Execution Time');disp(comp_time);
%End of Main Program

% Function KMmethod to compute IT2FS Centroid
%_____________________________________________

function centroid=KMmethod(LMF,UMF,x)

diff=ones(1,length(x))*5000;
for i=1:length(x)
   if UMF(i)>0
      theta(i,:)=[UMF(1:i) LMF(i+1:length(LMF))];
      centroid(i)=defuzz(x,theta(i,:),'centroid');
      diff(i)=abs(x(i)-centroid(i));
   end
end
[mindiff sw_index]=min(diff);
cl=x(sw_index);
theta_l=[UMF(1:sw_index) LMF(sw_index+1:length(LMF))];

diff=ones(1,length(x))*5000;
for i=1:length(x);
   if  i<length(x)
      if UMF(i+1)>0
         theta(i,:)=[LMF(1:i) UMF(i+1:length(LMF))];
         centroid(i)=defuzz(x,theta(i,:),'centroid');
         diff(i)=abs(x(i)-centroid(i));
      end
   end
end;
[mindiff sw_index]=min(diff);
cr=x(sw_index);
theta_r=[LMF(1:sw_index) UMF(sw_index+1:length(LMF))];
```

```
centroid=(cl+cr)/2;
end
% End of functiom KMmethod

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Error Metric Calculation

function [CFE,ME,MSE,RMSE,SD,MAD,MAPE]=errormetrics(predicted,TestCP)
% a=isnan(predicted1);
% p=size(TestCP1);
% z=1;
% for i=1:p
%   if a(i)~=1
%     predicted(z)=predicted1(i);
%     TestCP(z)=TestCP1(i);z=z+1;
%   end
% end
  CFE=sum(TestCP−predicted);
%   disp('CFE=');
%   disp(CFE);
  ME=mean(TestCP−predicted);
%   disp('ME=');
%   disp(ME);
  MSE=mean((TestCP−predicted).^2);
%   disp('MSE=');
%   disp(MSE);
  RMSE=sqrt(MSE);
%   disp('RMSE=');
%   disp(RMSE);
  SD=std(TestCP−predicted);
%   disp('SD=');
%   disp(SD);
  MAD=mean(abs(TestCP−predicted));
%   disp('MAD=');
%   disp(MAD);
  MAPE=mean(abs(TestCP−predicted)./TestCP)*100;
%   disp('MAPE=');
%   disp(MAPE);
End

%%%%%%%%%%%%%%%%%
% RMSE calculation

function rmse=evalfit(x,TestCP,TestA,UMF,LMF,typeMF,FLRG,WBS,TestB)
  Al=x(1);
```

```
  Au=x(end)+1;
        forecasted=predict(TestCP,TestA,Al,Au,UMF,LMF,typeMF,FLRG,WBS,
TestB);
  [~,~,~,rmse,~,~,~]=errormetrics(forecasted,TestCP);
End

%%%%%%%%%%%%%%%%%%
% Find Standard Deviations(SD)

function sd=extractSD(A,close)
  numFS=unique(A); % converting the close series into a matrix
  FS=zeros(length(numFS),length(close));
  for i=1:length(A)
    FS(A(i),i)=close(i);
  end
  b=1;
  for i=1:size(FS,1)
    temp=find(FS(i,:));
    %Case2: Partition has one point
    if length(temp)==1
      sd(b)=0.001;
%        flag(b)=2;
      b=b+1;
    %Case3: Partition has two points
    elseif length(temp)==2
      sd(b)=std(close(temp));
%        flag(b)=3;
      b=b+1;
    %Case 1 and 4: More than 2 contiguous or discrete points
    else
      indx=zeros(length(temp),1);
      l=1;
      for j=2:length(temp)  %contiguous points have been labelled sequen-
tially
        if (temp(j)-temp(j-1))==1
          indx(j-1)=l;
          indx(j)=l;
       elseif j>2 && (temp(j-1)-temp(j-2))==1 && (temp(j)-temp(j-1))~=1
          l=l+1;
        end
      end
      if max(indx)==0
```

```
        sd(b)=std(close(temp));
%         flag(b)=4;
        b=b+1;
      else
        for j=1:max(indx)
              temp1=temp(indx==j); % selecting days where the contigu-
ous points occur
          tempsd=std(close(temp1));
%           if tempsd>=1
            sd(b)=tempsd;
%             flag(b)=1;
            b=b+1;
%           end
        end
      end
    end
  end
end
%%%%%%%%%%%%%%%%%
% Forming Membership Functions(MFs)

function [UMF,LMF,typeMF]=formingMF(A,close,x,sd)
  numFS=unique(A); % converting the close series into a matrix
  FS=zeros(length(numFS),length(close));
  for i=1:length(A)
    FS(A(i),i)=close(i);
  end
  b=1;
  UMF=zeros(length(numFS),length(x));
  LMF=ones(length(numFS),length(x));
  typeMF=zeros(length(numFS),1);
  for i=1:size(FS,1)
    temp=find(FS(i,:));
    %Case2: Partition has one point
    if length(temp)==1
      m=round(close(temp));
      typeMF(i)=1;
      UMF(i,:)=max(UMF(i,:),trimf(x,[m-3*sd(b) m m+3*sd(b)]));
      LMF(i,:)=min(LMF(i,:),trimf(x,[m-3*sd(b) m m+3*sd(b)]));
      b=b+1;

    %Case3: Partition has two points
    elseif length(temp)==2
      m=round(mean(close(temp)));
```

```
      typeMF(i)=1;
      UMF(i,:)=max(UMF(i,:),trimf(x,[m−3*sd(b) m m+3*sd(b)]));
      LMF(i,:)=min(LMF(i,:),trimf(x,[m−3*sd(b) m m+3*sd(b)]));
      b=b+1;
    %Case 1 and 4: More than 2 contiguous or discrete points
    else
      indx=zeros(length(temp),1);
      l=1;
      for j=2:length(temp)  %contiguous points have been labelled sequen-
tially
        if (temp(j)−temp(j−1))==1
          indx(j−1)=l;
          indx(j)=l;
         elseif j>2 && (temp(j−1)−temp(j−2))==1 && (temp(j)−temp(j−1))~=1
          l=l+1;
        end
      end
      if max(indx)==0
        m=round(mean(close(temp)));
        typeMF(i)=1;
        UMF(i,:)=max(UMF(i,:),trimf(x,[m−3*sd(b) m m+3*sd(b)]));
        LMF(i,:)=min(LMF(i,:),trimf(x,[m−3*sd(b) m m+3*sd(b)]));
        b=b+1;
      else
%          c=0;
        for j=1:max(indx)
              temp1=temp(indx==j); % selecting days where the contigu-
ous points occur
          m=round(mean(close(temp1)));
%            if sd>=1
             UMF(i,:)=max(UMF(i,:),trimf(x,[m−3*sd(b) m m+3*sd(b)]));
             LMF(i,:)=min(LMF(i,:),trimf(x,[m−3*sd(b) m m+3*sd(b)]));
             b=b+1;
%              c=c+1;
%            end
        end
%          if c==1
%            typeMF(i)=1;
%          else
          typeMF(i)=2;
%          end
      end
    end
  end
```

```
  %Ensuring flat top
  loc1=0;loc2=0;
  for i=1:size(UMF,1)
    for j=1:1:size(UMF,2)
      if UMF(i,j)>0.999
        loc1=j;
        break;
      end
    end
    if j~=size(UMF,2)
      for j=size(UMF,2):-1:1
        if UMF(i,j)>0.999
          loc2=j;
          break;
        end
      end
    end
    if loc1~=0
      UMF(i,loc1:loc2)=1;
    end
  end
end
%%%%%%%%%%%%%%%%%%%%%%
% Gaussian MF Creation

function [UMF,LMF,typeMF]=formingMF(A,close,x,sd)
  numFS=unique(A); % converting the close series into a matrix
  FS=zeros(length(numFS),length(close));
  for i=1:length(A)
    FS(A(i),i)=close(i);
  end
  b=1;
  UMF=zeros(length(numFS),length(x));
  LMF=ones(length(numFS),length(x));
  typeMF=zeros(length(numFS),1);
  for i=1:size(FS,1)
    temp=find(FS(i,:));
    %Case2: Partition has one point
    if length(temp)==1
      m=round(close(temp));
      typeMF(i)=1;
      UMF(i,:)=max(UMF(i,:),gaussmf(x,[sd(b) m]));
      LMF(i,:)=min(LMF(i,:),gaussmf(x,[sd(b) m]));
      b=b+1;
```

```
    %Case3: Partition has two points
    elseif length(temp)==2
      m=round(mean(close(temp)));
      typeMF(i)=1;
      UMF(i,:)=max(UMF(i,:),gaussmf(x,[sd(b) m]));
      LMF(i,:)=min(LMF(i,:),gaussmf(x,[sd(b) m]));
      b=b+1;
    %Case 1 and 4: More than 2 contiguous or discrete points
    else
      indx=zeros(length(temp),1);
      l=1;
      for j=2:length(temp)  %contiguous points have been labelled sequen-
tially
        if (temp(j)−temp(j−1))==1
          indx(j−1)=l;
          indx(j)=l;
       elseif j>2 && (temp(j−1)−temp(j−2))==1 && (temp(j)−temp(j−1))∼=1
          l=l+1;
        end
      end
      if max(indx)==0
        m=round(mean(close(temp)));
        typeMF(i)=1;
        UMF(i,:)=max(UMF(i,:),gaussmf(x,[sd(b) m]));
        LMF(i,:)=min(LMF(i,:),gaussmf(x,[sd(b) m]));
        b=b+1;
      else
%          c=0;
        for j=1:max(indx)
              temp1=temp(indx==j); % selecting days where the contigu-
ous points occur
           m=round(mean(close(temp1)));
%            if sd>=1
             UMF(i,:)=max(UMF(i,:),gaussmf(x,[sd(b) m]));
             LMF(i,:)=min(LMF(i,:),gaussmf(x,[sd(b) m]));
             b=b+1;
%              c=c+1;
%            end
        end
%          if c==1
%            typeMF(i)=1;
%          else
           typeMF(i)=2;
%          end
```

```
      end
    end
  end
  %Ensuring flat top
  loc1=0;loc2=0;
  for i=1:size(UMF,1)
    for j=1:1:size(UMF,2)
      if UMF(i,j)>0.999
        loc1=j;
        break;
      end
    end
    if j~=size(UMF,2)
      for j=size(UMF,2):-1:1
        if UMF(i,j)>0.999
          loc2=j;
          break;
        end
      end
    end
    if loc1~=0
      UMF(i,loc1:loc2)=1;
    end
  end
end
%%%%%%%%%%%%%%
% FUZZY Secondary Factor Variation


function BSF=fuzzyvarSF(VarSF)
BSF=zeros(length(VarSF),1);
for i=2:length(VarSF)
  for j=1:14
    if VarSF(i)<-6
      BSF(i)=1;
    elseif VarSF(i)>=6
      BSF(i)=14;
    elseif VarSF(i)>=(j-1)-6 && VarSF(i)<j-6
      BSF(i)=j+1;
    end
  end
end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
%% MF Using DE

function     [UMF,LMF,typeMF,rmse]=MFusingDE(A,close,x,sd,TestCP,TestA,
FLRG,WBS,TestB)
  genmax=50;
  F=0.2; % Scale Factor
  Cr=0.9; % Cross-over probability
  NP=20; % no. of population members
  gen=1;
  %% Initialization
  Zmin=ones(1,length(sd))*0.1;
  Zmax=sd;
  Z=zeros(NP,length(sd));
  for i=1:NP
    Z(i,:)=Zmin+rand*(Zmax-Zmin);
  end
  %%
  rmse=zeros(genmax,NP);
  while(gen<=genmax)
    disp('Gen=');
    disp(gen);
    %% Mutation
    V=zeros(NP,length(sd));
    for i=1:NP
      j=datasample(1:NP,1);
      while j==i
        j=datasample(1:NP,1);
      end
      k=datasample(1:NP,1);
      while k==i || k==j
        k=datasample(1:NP,1);
      end
      l=datasample(1:NP,1);
      while l==i || l==j || l==k
        l=datasample(1:NP,1);
      end
      V(i,:)=Z(j,:)+F.*(Z(k,:)-Z(l,:));
      for j=1:length(sd) % Ensuring V(i,j) is within Zmax and Zmin
        if V(i,j)<Zmin(j)
          V(i,j)=Zmin(j);
        elseif V(i,j)>Zmax(j)
          V(i,j)=Zmax(j);
        end
      end
```

```
    end
    %% Crossover
    U=zeros(NP,length(sd));
    for i=1:NP
      for j=1:length(sd)
        if rand<=Cr
          U(i,j)=V(i,j);
        else
          U(i,j)=Z(i,j);
        end
      end
    end
    %% Selection
    for i=1:NP
    [UMFu,LMFu,typeMFu]=formingMF(A,close,x,U(i,:));
    [UMFz,LMFz,typeMFz]=formingMF(A,close,x,Z(i,:));
      if evalfit(x,TestCP,TestA,UMFu,LMFu,typeMFu,FLRG,WBS...
          ,TestB)<evalfit(x,TestCP,TestA,UMFz,LMFz,typeMFz...
          ,FLRG,WBS,TestB)
        Z(i,:)=U(i,:);
      end
    end
    %% Storing fitness over generations
    for i=1:NP
      [UMFz,LMFz,typeMFz]=formingMF(A,close,x,Z(i,:));
      rmse(gen,i)=evalfit(x,TestCP,TestA,UMFz,LMFz,typeMFz...
        ,FLRG,WBS,TestB);
    end
    %%
    gen=gen+1;
  end
  [~,indx]=min(rmse(genmax,:));
  [UMF,LMF,typeMF]=formingMF(A,close,x,Z(indx,:));
End
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Over All Variation

function VarSF=overallvar(VarMF,SF,e,n)
  %
  VarSF1=zeros(size(SF,1),n);
  for i=1:n
    for j=2:size(SF,1)
      VarSF1(j,i)=(SF(j,i)-SF(j-1,i))*100/SF(j-1,i);
    end
  end
```

```
  %
  %
  tempDiffer=zeros(e,n);
  for i=1:n
    for j=3:e
      tempDiffer(j,i)=abs(VarSF1(j−1,i)−VarMF(j));
    end
  end
  DifferSF=sum(tempDiffer);
  %
  %
  WVSF=zeros(1,n);
  for i=1:n
    WVSF(i)=sum(DifferSF)/DifferSF(i);
  end
  %
  %
  WSF=zeros(1,n);
  for i=1:n
    WSF(i)=WVSF(i)/sum(WVSF);
  end
  %
  %
  VarSF=zeros(1,size(SF,1));
  for i=1:size(SF,1)
    VarSF(i)=sum(VarSF1(i,:).*WSF);
  end
  %
End
%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Partitioning The Universe of Discourse(UOD)



function [A,B,Var,Au,Al]=partitioning(CP)
%%
A=zeros(length(CP),1);
B=zeros(length(CP),1);
Var=zeros(length(CP),1);
%%
```

```
templ=min(CP);
tempu=max(CP);

if (roundn(templ,2)−templ)>0
  Al=roundn(templ,2)−100;
else
  Al=roundn(templ,2);
end

% if (roundn(tempu,2)−tempu)>0
%   Au=roundn(tempu,2);
% else
%   Au=roundn(tempu,2)+200;
% end
Au=Al;
while(1)
  Au=Au+200;
  if Au>tempu
    break;
  end
end

nFS=(Au–Al)/200;

for i=1:length(CP) %partioning the close series of main factor
  for j=1:nFS
    if CP(i)>=(j−1)*200+Al && CP(i)<j*200+Al
      A(i,1)=j;
      break;
    end
  end
end
%%
for i=2:length(CP) % finding the var series
    Var(i)=(CP(i)−CP(i−1))*100/CP(i−1);
end

for i=2:length(Var) % partioning the var series
  for j=1:14
    if Var(i)<−6
      B(i)=1;
    elseif Var(i)>=6
      B(i)=14;
    elseif Var(i)>=(j−1)−6 && Var(i)<j−6
      B(i)=j+1;
```

```
      end
    end
  end
  end
  %%%%%%%%%%%%%%%%%%%%%%%%%
  %% Plot RMSE with adaptation

  function plotDE(rmse)
    figure,plot(1:size(rmse,1),min(rmse,[],2),'kx-','MarkerSize',5);
    xlabel('Generations -->');
    ylabel('RMSE -->');
    title('Evolving parameters to minimize RMSE');
  end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %% Plot Forecasted Price

  function plotforecasted(predicted,TestCP,RMSE)
  figure,
  subplot(3,2,[1 2 3 4]);
  plot(TestCP,'k*:');
  hold on
  plot(predicted,'ko-');
  ylabel('Close Price');
  axis([0 length(TestCP)+5 min(min(predicted),min(TestCP))-1000 max(max
  (predicted),max(TestCP))+1000]);
  hold off
  legend('Actual','Predicted','location','SouthEast');
  subplot(3,2,[5 6]);
  stem(TestCP-predicted,'ko-');
  text(27,max(TestCP-predicted)+50,{'RMSE=',RMSE});
  axis([0 length(TestCP) min(TestCP-predicted) max(TestCP-predicted)
  +150]);
  xlabel('Testing days');
  ylabel('Error');
  end
  %%%%%%%%%%%%%%%%%%%%%%%%%
  %% Plot FOU(Foot Print of Uncertainity)

  function plotFOU(UMF,LMF,x)
    for i=1:size(UMF,1)
      figure,shadedplot(x,LMF(i,:),UMF(i,:),[0.8 0.8 0.8]);
      xlabel('Close');
      ylabel('Membership values');
      title(['FOU for fuzzy set A', num2str(i)]);
    end
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Plot Partitions

function plotpartitions(close,A,Al,Au)
   numFS=unique(A);  % converting the close series into a matrix
   FS=zeros(length(numFS),length(close));
   for i=1:length(A)
      FS(A(i),i)=close(i);
   end

   % Plot Input Close
   plot(close,'k*−');
   hold on;
   for i=1:length(numFS)
      part=Al+(i−1)*200;
      plot([1 length(close)],[part part],'k:');
      hold on;
   end
   hold off;
   axis([1 length(close) Al Au]);
   xlabel('Training days');
   ylabel('Close');
   title('Fuzzifying training data');
   %
End
%%%%%%%%%%%%%%%%%%%%%%
%% PredictionFunction

function forecasted=predict(TestCP,A,Al,Au,UMF,LMF,typeMF,FLRG,WBS,B
x=Al:Au−1;
AB=horzcat(A,B);
forecasted=zeros(length(TestCP),1);
for i=2:size(AB,1)
   a=AB(i−1,1);
   b=AB(i−1,2);
   consequent=find(FLRG(a,:,b));
   centroid=zeros(1,size(FLRG,1));
   if isempty(consequent)
      if typeMF(a)==1
         forecasted(i−1)=sum(x.*UMF(a,:))/sum(UMF(a,:));
      elseif typeMF(a)==2
         avgMF=(LMF(a,:)+UMF(a,:))/2;
         forecasted(i−1)=sum(x.*avgMF)/sum(avgMF);
%          forecasted(i−1)=KMmethod(LMF(a,:),UMF(a,:),x);
```

```
    end
  else
    %SF1=0;SF2=0;SF3=0;
    NF=0;
    for j=1:length(consequent)
      %Case1: a=T1FS, consequent(j)=T1FS
      if typeMF(a)==1 && typeMF(consequent(j))==1
        predy=interp1(x,UMF(a,:),TestCP(i−1),'linear','extrap');
        temp=ones(1,length(x))*predy;
        projMF=min(temp,UMF(consequent(j),:));
        centroid(consequent(j))=sum(x.*projMF)/sum(projMF);
      %Case2: a=T1FS, consequent(j)=IT2FS
      elseif typeMF(a)==1 && typeMF(consequent(j))==2
        predy=interp1(x,UMF(a,:),TestCP(i−1),'linear','extrap');
        temp=ones(1,length(x))*predy;
        projUMF=min(temp,UMF(consequent(j),:));
        projLMF=min(temp,LMF(consequent(j),:));
        avgMF=(projLMF+projUMF)/2;
        centroid(consequent(j))=sum(x.*avgMF)/sum(avgMF);
%          centroid(consequent(j))=KMmethod(projLMF,projUMF,x);
      %Case3: a=IT2FS, consequent(j)=T1FS
      elseif typeMF(a)==2 && typeMF(consequent(j))==1
        predU=interp1(x,UMF(a,:),TestCP(i−1),'linear','extrap');
%          predL=interp1(x,LMF(a,:),TestCP(i−1),'linear','extrap');
        temp=ones(1,length(x))*predU;
        projMF=min(temp,UMF(consequent(j),:));
        centroid(consequent(j))=sum(x.*projMF)/sum(projMF);
      %Case4: a=IT2FS, consequent(j)=IT2FS
      elseif typeMF(a)==2 && typeMF(consequent(j))==2
        predU=interp1(x,UMF(a,:),TestCP(i−1),'linear','extrap');
        predL=interp1(x,LMF(a,:),TestCP(i−1),'linear','extrap');
        tempU=ones(1,length(x))*predU;
        tempL=ones(1,length(x))*predL;
        projUMF=min(tempU,UMF(consequent(j),:));
        projLMF=min(tempL,LMF(consequent(j),:));
        avgMF=(projLMF+projUMF)/2;
        centroid(consequent(j))=sum(x.*avgMF)/sum(avgMF);
%          centroid(consequent(j))=KMmethod(projLMF,projUMF,x);
      end
      if j<a
         forecasted(i−1)=forecasted(i−1)+centroid(consequent(j))*FLRG
(a,consequent(j),b)*WBS(b,1);
      elseif j==a
         forecasted(i−1)=forecasted(i−1)+centroid(consequent(j))*FLRG
(a,consequent(j),b)*WBS(b,2);
```

```
      else
         forecasted(i−1)=forecasted(i−1)+centroid(consequent(j))*FLRG
(a,consequent(j),b)*WBS(b,3);
      end
    end
    for j=1:length(consequent)
      if j<a
        NF=NF+FLRG(a,consequent(j),b)*WBS(b,1);
      elseif j==a
        NF=NF+FLRG(a,consequent(j),b)*WBS(b,2);
      else
        NF=NF+FLRG(a,consequent(j),b)*WBS(b,3);
      end
    end
    forecasted(i−1)=forecasted(i−1)/NF;
  end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%
%% Shade Plot

function [ha hb hc] = shadedplot(x, y1, y2, varargin)
y = [y1; (y2−y1)]';
ha = area(x, y);
set(ha(1), 'FaceColor', 'none') % this makes the bottom area invisible
set(ha, 'LineStyle', 'none')

% plot the line edges
hold on
hb = plot(x, y1, 'k', 'LineWidth', 2);
hc = plot(x, y2, 'k', 'LineWidth', 2);
hold off

% set the line and area colors if they are specified
switch length(varargin)
  case 0
  case 1
    set(ha(2), 'FaceColor', varargin{1})
  case 2
    set(ha(2), 'FaceColor', varargin{1})
    set(hb, 'Color', varargin{2})
    set(hc, 'Color', varargin{2})
  otherwise
end
```

```
% put the grid on top of the colored area
set(gca, 'Layer', 'top')
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Table IX

function FVG=tableIX(B_MF,B_SF)

  FVG=zeros(14,14);
  temp=zeros(14,14);

  for t=2:length(B_MF)
    Bx=B_MF(t);
    Bz=B_SF(t-1);
    temp(Bz,Bx)=1;
    FVG=FVG+temp;
    temp=zeros(14,14);
  end

end
%%%%%%%%%%%%%%%%%%%%%%%
%% TableVI

function FLRG=tableVI(B,fromA,toA)

  endA=max([max(fromA),max(toA)]);
  FLRG=zeros(endA,endA,14);

  temp=zeros(endA,endA,14);
  for i=1:length(B)
    temp(fromA(i),toA(i),B(i))=1;
    FLRG=FLRG+temp;
    temp=zeros(endA,endA,14);
  end

end
%%%%%%%%%%%%%%%%%%%%%%
%% Table X

function BS=tableX(FVG)
  BS=zeros(14,3);

  for i=1:14
    if sum(FVG(i,:))~=0
      BS(i,1)=sum(FVG(i,1:i-1))/sum(FVG(i,:));
```

```
      BS(i,2)=FVG(i,i)/sum(FVG(i,:));
      BS(i,3)=sum(FVG(i,i+1:14))/sum(FVG(i,:));
    end
  end
end
%%%%%%%%%%%%%%%%
```

How to run the program with workspace construction in MATLAB is available in the url: http://computationalintelligence.net/fuzzytimeseries/howtorun.html

References

1. Engineering Statistic and book [Online]. http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc41.htm
2. Chen, S. M., & Chen, C. D. (2011). Handling forecasting problems based on high-order fuzzy logical relationships. *Expert Systems with Applications, 38*(4), 3857–3864.
3. Chen, S. M., & Tanuwijaya, K. (2011). Fuzzy forecasting based on high-order fuzzy logical relationships and automatic clustering techniques. *Expert Systems with Applications, 38*(12), 15425–15437.
4. Wu, C. L., & Chau, K. W. (2013). Prediction of rainfall time series using modular soft computing methods. *Elsevier, Engineering Applications of Artificial Intelligence, 26,* 997–1007.
5. Wu, C. L., Chau, K. W., & Fan, C. (2010). Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques. *Elsevier, Journal of Hydrology, 389*(1–2), 146–167.
6. Barnea, O., Solow, A. R., & Stonea, L. (2006). On fitting a model to a population time series with missing values. *Israel Journal Of Ecology & Evolution, 52,* 1–10.
7. Chen, S. M., & Hwang, J. R. (2000). Temperature prediction using fuzzy time series. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics, 30*(2), 263–275.
8. Song, Q., & Chissom, B. S. (1993). Fuzzy time series and its model. *Fuzzy Sets and Systems, 54*(3), 269–277.
9. Song, Q., & Chissom, B. S. (1993). Forecasting enrollments with fuzzy time series—Part I. *Fuzzy Sets and Systems, 54*(1), 1–9.
10. Song, Q., & Chissom, B. S. (1994). Forecasting enrollments with fuzzy time series—Part II. *Fuzzy Sets and Systems, 62*(1), 1–8.
11. Song, Q. (2003). A note on fuzzy time series model selection with sample autocorrelation functions. *Cybernetics &Systems, 34*(2), 93–107.
12. Jalil, A., & Idrees, M. (2013). Modeling the impact of education on the economic growth: Evidence from aggregated and disaggregated time series data of Pakistan. *Elsevier Economic Modelling, 31,* 383–388.
13. Edwards, R. P., Magee, J., & Bassetti, W. H. C. (2007). *Technical analysis of Stock Trends, 9th Edition, p. 10*. New York, USA: Amacom.
14. Raynor, W. J., Jr. (1999) *The international dictionary of artificial intelligence*. IL: Glenlake Publishing.
15. Mallat, S. (1997). *A wavelet tour of signal processing*. San diego, CA, USA: Academic Press.
16. Bhowmik, P., Das, S., & Konar, A., Nandi, D., & Chakraborty, A. (2010) Emotion clustering from stimulated encephalographic signals using a Duffing Oscillator. *International Journal of Computers in Healthcare*, *1*(1).

17. Gupta, S. C., & Kapoor, V. K. (2002). *Fundamental of mathematical statistics*. New Delhi, India: S. Chand & Sons.
18. Jansen,B. H., Bourne, J. R., Ward, J. W. (1981). Autoregressive estimation of short segment spectra for computerized EEG analysis. *IEEE Transactions on Biomedical Engineering*, *28*(9).
19. Hjorth, B. (1970). EEG analysis based on time domain properties. *Electroencephalography and Clinical Neurophysiology, 29,* 306–310.
20. Sanei, S. (2013) *Adaptive processing of brain signals*. USA: Wiley.
21. Chen, S. M. (1996). Forecasting enrollments based on fuzzy time series. *Fuzzy Sets and Systems, 81*(3), 311–319.
22. Hwang, J. R., Chen, S. M., & Lee, C. H. (1998). Handling forecasting problems using fuzzy time series. *Fuzzy Sets and Systems, 100*(1–3), 217–228.
23. Lee, L. W., Wang, L. H., & Chen, S. M. (2007). Temperature prediction and TAIFEX forecasting based on fuzzy logical relationships and genetic algorithms. *Expert Systems with Applications, 33*(3), 539–550.
24. Wong, W. K., Bai, E., Chu, A. W. C. (2010). Adaptive time-variant models for fuzzy-time-series forecasting. *IEEE Transaction on Systems*, *Man*, *Cybernetics-Part B: Cybernetics*, *40*(6).
25. Cheng, C. H., Chen, T. L., Teoh, H. J., & Chiang, C. H. (2008). Fuzzy time-series based on adaptive expectation model for TAIEX forecasting. *Elsevier, Expert Systems with Applications, 34,* 1126–1132.
26. Chen, T. L., Cheng, C. H., & Teoh, H. J. (2007). Fuzzy time-series based on Fibonacci sequence for stock price forecasting. *Elsevier, Physica A, 380,* 377–390.
27. Huarng, K., & Yu, T. H. K. (2006). The application of neural networks to forecast fuzzy time series. *Physica A, 363*(2), 481–491.
28. Yu, T. H. K., & Huarng, K. H. (2010). A neural network-based fuzzy time series model to improve forecasting. *Expert Systems with Applications, 37*(4), 3366–3372.
29. Kuo, I. H., Horng, S. J., Kao, T. W., Lin, T. L., Lee, C. L., & Pan, Y. (2009). An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization. *Expert Systems with Applications, 36*(3), 6108–6117.
30. Lu, W., Yang, J., Liu, X., & Pedrycz, W. (2014). The modeling and prediction of time-series based on synergy of high order fuzzy cognitive map and fuzzy c-means clustering. *Elsevier, Knowledge Based Systems, 70,* 242–255.
31. Yu, T. H. K., & Huarng, K. H. (2008). A bivariate fuzzy time series model to forecast the TAIEX. *Expert Systems with Applications, 34*(4), 2945–2952.
32. Yu, T. H. K., & Huarng, K. H. (2010). Corrigendum to "A bivariate fuzzy time series model to forecast the TAIEX". *Expert Systems with Applications, 37*(7), 5529.
33. Chen, S. M., & Chang, Y. C. (2010). Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques. *Information Sciences, 180*(24), 4772–4783.
34. Wong,H. L., Tu, Y. H., & Wang, C. C. (2009). An evaluation comparison between multivariate fuzzy time series with traditional time series model for forecasting Taiwan export. In *Proceedings of World Congress on Computer Science Engineering*, pp. 462–467.
35. Chen, S. M., & Tanuwijaya, K. (2011). Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering clustering techniques. *Expert Systems with Applications, 38* (8), 10594–10605.
36. Huarng, K., Yu, H. K., & Hsu, Y. W. (2007). A multivariate heuristic model for fuzzy time-series forecasting. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics, 37*(4), 836–846.
37. Sun, B. Q., Guo, H., Karimi, H. R., Ge, Y., Xiong, S. (2015). Prediction of stock index futures prices based on fuzzy sets and multivariate fuzzy time series. *Neurocomputing*, *Elsevier*, *151*, 1528–1536.
38. Chen, T. L., Cheng, C. H., & Teoh, H. J. (2008). High-order fuzzy time-series based on multi-period adaptation model for forecasting stock markets. *Elsevier, Physica A, 387,* 876–888.

39. Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8,* 338–353.
40. Zhang, W. (1992). *Applications engineer*, Aptronics Incorporated, Copyright © by Aptronix Inc.
41. Virkhareg, N., & Jasutkar, R. W. (2014) Neuro-fuzzy controller based washing machine. *International Journal of Engineering Science Invention*, *3*, 48–51.
42. Konstantinidis, K., Gasteratos, A., & Andreadis, I. (2005). Image retrieval based on fuzzy color histogram processing. *Elsevier, Optics Communications, 248,* 375–386.
43. Halder, A., Konar, A., Mandal, R., Chakraborty, A., Bhowmik, P., Pal, N. R., et al., (2013) General and interval type-2 fuzzy face-space approach to emotion recognition. *IEEE Transactions On Systems*, *Man*, *And Cybernetics: Systems*, *43*(3).
44. Karnik, N. N., & Mendel, J. M. (1999). Applications of type-2 fuzzy logic systems to forecasting of time-series. *Elsevier, Information Sciences, 120,* 89–111.
45. Huarng, K., & Yu, H. K. (2005). A type 2 fuzzy time series model for stock index forecasting. *Physica A, 353,* 445–462.
46. Bajestani, N. S., & Zare, A. (2011). Forecasting TAIEX using improved type 2 fuzzy time series. *Expert Systems with Applications, 38*(5), 5816–5821.
47. Chen, S. M., & Chen, C. D. (2011). TAIEX forecasting based on fuzzy time series and fuzzy variation groups. *IEEE Transactions on Fuzzy Systems, 19*(1), 1–12.
48. Mendel, J. M., & Wu, D. (2010). *Perceptual computing: Aiding people in making subjective judgements*. Hoboken, NJ: IEEE-Wiley press.
49. Chakraborty, S., Konar, A., Ralescu, A., & Pal, N. R. (2015). A fast algorithm to compute precise type-2 centroids for real time control applications. *IEEE Transaction on Cybernetics, 45*(2), 340–353.
50. Chen,S. M., Chu, H. P., Sheu, T. W. (2012). TAIEX forecasting using fuzzy time series and automatically generated weights of multiple factors. *IEEE Transaction on Systems*, *Man*, *and Cybernetics-Part A: Systems and Humans*, *42*(6), 1485–1495.
51. Mendel,J. M., Hagras, H., Tan, W. W., Melek, W. W., & Ying, H. (2014) *Introduction to type-2 fuzzy logic control: Theory and applications*. USA: IEEE-Wiley Press.
52. Nilsson, N. J. (1980). *Principles ofartificial intelligence*. CA: Morgan Kaufmann.
53. Web Link: http://www.computationalintelligence.net/ieee_cyb/index_prediction_appendix1.pdf
54. Garibaldi,J. M. (2010). Alternative forms of non-standard fuzzy sets: A discussion chapter. In *Proceedings of Type-2 Fuzzy Logic: State of the Art and Future Directions*, London.
55. Wu, D. (2011). A constraint representation theorem for interval type-2 fuzzy sets using convex and normal embedded type-1 fuzzy sets, and its application to centroid computation. In *Proceedings of World Conference on Soft Computing*, San Francisco, CA.
56. Web Link for historical Data TAIEX. [Online]. Available: http://www.twse.com.tw/en/products/indices/tsec/taiex.php
57. Storn, R., & Price, K. G. (1997). Differential Evolution- a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359.
58. Das, S., & Suganthan, P. N. (2011) Differential evolution: A survey of the state-of-the-art. *IEEE Transaction on Evolutionary Computation*, *15*(1), 4–31.
59. Chen, S. M., Kao, P.-Y. (2013). TAIEX forecasting based on time series, particle swarm optimization techniques and support vector machines. *Information Science*, *Elsevier*, *247*, 62–71.
60. Cai, Q., Zhang, D., Zheng, W., & Leung, S. C. H. (2015). A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression. *Knowledge-Based Systems, 74,* 61–68.
61. Chen, Mu-Yen. (2014). A high-order fuzzy time series forecasting model for internet stock trading. *Future Generation Computer Systems Elsevier, 37,* 461–467.
62. Gardner, E., & McKenzie, E. (1989). Seasonal exponential smoothing with damped trends. *Management Science, 35*(3), 372–376.

63. Kuremoto, T., Kimura, S., Kobayashi, K., & Obayashi, M. (2014). Time series forecasting using a deep beliefnetwork with restricted boltzmann machines (online open access). *Neurocomputing, 137*(5), 47–56.
64. Yildiz,O. T. (2013) Omnivariate rule induction using a novel pairwise statistical test. *IEEE Transactions On Knowledge And Data Engineering*, *25*(9).

http://www.springer.com/978-3-319-54596-7

Time-Series Prediction and Applications
A Machine Intelligence Approach
Konar, A.; Bhattacharya, D.
2017, XVIII, 242 p. 69 illus., 13 illus. in color., Hardcover
ISBN: 978-3-319-54596-7