Knowledge Extraction from a Time-Series Using Segmentation, Fuzzy Matching and Predictor Graphs

Jishnu Mukhoti[1], Pratyusha Rakshit[2], Diptendu Bhattacharya[3], Amit Konar[4]
[1]Department of Computer Science and Engineering,
[2,3,4]Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata, India,
[1]omegafragger@gmail.com, [2]pratyushar1@gmail.com,
[3]diptendu1@gmail.com, [4]konaramit@yahoo.co.in

Atulya K. Nagar
Department of Math and Computer Science,
Liverpool Hope University,
Liverpool, U.K
nagara@hope.ac.uk

***Abstract*— In this paper, a novel multi-stage approach to knowledge extraction from a time-series is proposed. A given time-series is modeled as a sequence of well-known primitive patterns with the purpose of identifying first-order probabilistic transition rules for prediction. The first stage of the proposed model segments a time-series into structurally distinct temporal blocks of non-uniform length such that each block possesses a relatively low variation of dynamic slope. In the second stage, the temporal segments thus obtained are normalized and matched with one of four well-known primitive patterns using a fuzzy matching algorithm. Finally, the sequence of matched segments is used to represent the time-series as a set of four directed graphs corresponding to the four primitive patterns. Each vertex in the graphs represents a horizontal partition of the time-series and each directed edge indicates the transitions between two such partitions caused by the occurrence of one or more temporal segments. In the test phase, the graphs are employed to predict possible future values of the time-series. Experiments carried out on the TAIEX close-price time-series indicate a high prediction accuracy, thereby validating the use of the model for real-life forecasting applications.**

Keywords*— *Knowledge extraction, time-series segmentation, fuzzy matching, directed graph.

I. Introduction

A time-series is a discrete sequence of real-valued observations obtained by sampling a measurable phenomenon at regular intervals of time. Accurate prediction of future values of a given time-series has been a well-known topic for research. However, the behavior of a time-series is often dependent on multiple factors, many of which are unknown and erratic in nature. Therefore, the general approach employed for time-series analysis is to develop a mathematical model of the series and to make predictions based on the model. Quite a few approaches have been developed in this direction.

Among the well-known methodologies that have found their way into time-series analysis, the application of fuzzy set theory [1] is worth special mention. The fuzzy time-series was first defined by Song and Chissom in their works [2]-[4]. The general approach proposed by them was to consider the dynamic range of the time-series as a universe of discourse and to construct fuzzy sets on disjoint contiguous intervals, called partitions defined on the universe. The time-series is then fuzzified by replacing each data point by the fuzzy set to which it belongs with highest membership value. The fuzzified time-series can be utilized to derive first and higher order fuzzy logical relations (FLRs) in order to make predictions on possible future values of the time-series.

Several improvements have been proposed on the original model of Song and Chissom. In [5], the authors proposed to weight the FLRs based on the chronological order of their occurrence, thereby taking into account the recurrence of FLRs. The incorporation of secondary and multifactor heuristics in the modeling of a fuzzy time-series was proposed in [6] and extended in [7], [8] for better prediction accuracy. Other interesting works in this regard include efficient time-series partitioning and fuzzification schemes [9], [10], application of optimization techniques like PSO [11] and ant-colony optimization [12], fuzzy variation groups [13] and others.

In the current paper, we model a time-series as a sequence of well-known primitive patterns occurring in contiguous time-slots. The proposed approach has three primary stages. The first stage deals with semantic segmentation of a time-series into disjoint temporal blocks of non-uniform length such that each segment can be efficiently categorized into one of the known primitive patterns. The proposed segmentation algorithm identifies temporal regions where the dynamic slope of the time-series possesses a high variance. Such temporal regions are then marked as segment boundaries to indicate a structural change in the time-series. The advantages of the segmentation algorithm lie primarily in its simplicity and low computational overhead.

The time-segments thus obtained in the previous step are normalized and represented as a sequence of 10 points to achieve uniformity in their respective lengths. The second stage involves matching the segments with one of four well-known patterns, namely, the linear rise, the linear fall, the Gaussian bell curve and the inverse Gaussian curve (or the inverted bell). A fuzzy matching algorithm is employed to derive a similarity metric between a normalized segment and a primitive pattern. This provides for real-time approximate classification of segments without the need to train any computational model.

978-1-5090-0626-7/16/$31.00

In the third stage, the information obtained from the previous steps is used to represent the time-series in the form of a set of weighted directed graphs where each graph corresponds to a primitive pattern. The nodes in the graph for each pattern *P* represent horizontal partitions in the time-series and the weighted directed edge between two nodes denotes the probability of occurrence of transitions between the two corresponding partitions caused by one or more time-segments classified to pattern *P*. The graphs can be utilized to derive first or higher order probabilistic transition rules as well as make predictions based on such derived rules. Hence, the graphs are called predictor graphs.

Experiments carried out on the TAIEX [14] economic close-price time-series for the period 1990-1999 indicate a low average RMSE of 88.48 outperforming competitive models. Figure 1 provides a schematic block diagram for the proposed model.

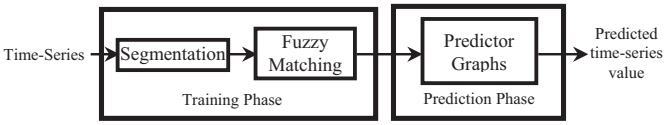

Fig. 1. Schematic block-diagram of proposed model

The rest of the paper is organized as follows. Section II deals with a brief overview of fuzzy sets and time-series partitioning. Section III discusses the proposed segmentation algorithm. The fuzzy matching algorithm is described in Section IV. Section V deals the formation of predictor graphs. Experiments and performance analysis of the proposed model are discussed in Section VI. Conclusions are listed in Section VII.

II. Preliminaries

In this section, we provide an overview of a few concepts required to understand the remainder of this paper. They include a brief description of fuzzy sets and fuzzy membership functions as well as concepts related to partitioning of a time-series.

A. Fuzzy Sets

Definition 1: A **fuzzy set** *A* defined on a universe of discourse *U* is a set of 2-tuples as follows:

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (1)$$

where x is an element of the universe U and $\mu_A(x) \in [0,1]$ represents the membership value of x in the fuzzy set A. In simple terms, a fuzzy set is a generalization of a conventional set where an element may belong with a membership value lying in the range [0,1]. On the other hand, an element belongs to a conventional set with a membership value of either 1 (member of the set) or 0 (not a member of the set).

Definition 2: For a given fuzzy set A defined on a universe of discourse U, a function of the form $\mu_A : U \rightarrow [0,1]$ which maps each element $x \in U$ to its corresponding membership value $\mu_A(x) \in [0,1]$ in the fuzzy set A is called a **fuzzy membership function**. Such a membership function is often required when the universe of discourse is a continuous (or infinite) set, thereby making it impossible to define a finite number of 2-tuples for a fuzzy set.

Fuzzy sets are generally employed to describe collections of objects which are not precisely defined. For instance, a set of very tall people or a set of numbers which are very close to 0 can be defined using fuzzy sets. The linguistic terms "very tall" and "very close to 0" are subjective and dependent on individual judgment and hence, conventional sets cannot be used to properly define such collections.

B. Time-Series Partitioning

Definition 3: A **time-series** is defined as a sequence of discrete values or measurements obtained by sampling a measurable phenomenon at successive points in time. Let y be some measurable entity and let $y(t)$ be the sampled value of the entity obtained at time-instant t. Then the time-series $\vec{C}$ of length n can be defined as the vector

$$\vec{C} = [y(T),\ y(2T),\ ...,\ y(nT)] = [c_1,\ c_2,\ ...,\ c_n] \quad (2)$$

where n is the number of samples obtained and T is the sampling interval.

Definition 4: Let $R(\vec{C}) = [c_{\min}, c_{\max}]$ denote the dynamic range of the time-series $\vec{C}$ where $c_{\min}$ and $c_{\max}$ are the minimum and maximum values of the series respectively. Dividing the range $R(\vec{C})$ into k non-overlapping, contiguous intervals of the form $P_1 = [p_1^-, p_1^+],\ P_2 = [p_2^-, p_2^+],\ ...,\ P_k = [p_k^-, p_k^+]$ where

$$p_i^- > p_{i-1}^+,\ \forall i \in \{2,3,...,k\} \quad (3)$$

is called **partitioning** and each such interval P_j is called a **partition**. Clearly, $p_1^- = c_{\min}$ and $p_k^+ = c_{\max}$. Figure 2 shows a time-series partitioned into 7 equi-spaced partitions.

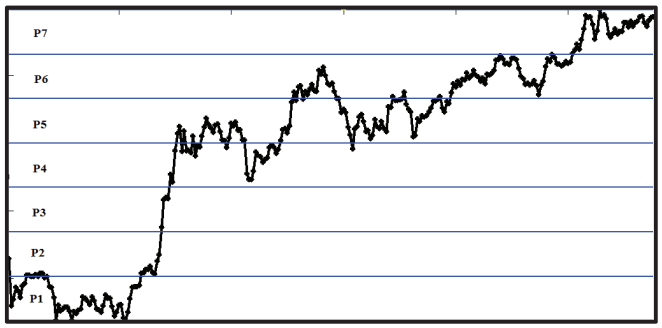

Fig. 2. Partitioning a sample time-series into 7 equi-spaced partitions P1 to P7

III. Edge-Profile Time-Series Segmentation (EPTS)

In this section, we propose a technique for efficient time-series segmentation by incorporating the variation of dynamic slope of the time-series as the primary segmentation criterion. The idea behind semantic segmentation of a time-series is to identify time points where the series displays a drastic change in slope. In the proposed approach, the time-series is viewed as a piecewise linear curve where each pair of consecutive data points in the series is joined by a straight line, called an edge. Let us consider an ideal time-series without the presence of low level noisy fluctuations. The slopes of the edges joining consecutive data points show a high variation only in regions where the series displays a change in its behavioral pattern. This intuition is applied to our approach where we identify such temporal regions and mark them as segment boundaries in the time-series. The data points of the series lying between two consecutive segment boundaries is called a temporal segment or a time block. Since,

the segmentation criterion is based on the relative variance of edge slopes in the series, the proposed technique is named Edge Profile Time-Series Segmentation (EPTS). The advantage of the algorithm lies primarily in its simplicity and low computational complexity with respect to other existing time-series segmentation algorithms. The following definitions formally explain the proposed approach.

A. Definitions

Definition 5: Let c_i and c_{i+1} be two consecutive data points of a given time-series $\vec{C}$, sampled at time-instants iT and $(i+1)T$ respectively. Then, the straight line $l_{1,2}$ formed between the points $p_1 \equiv (iT, c_i)$ and $p_2 \equiv ((i+1)T, c_{i+1})$ is called an **edge** of the time-series and the slope of the edge is given by

$$slope(p_1, p_2) = \frac{(c_{i+1} - c_i)}{((i+1)T - iT)} = \frac{(c_{i+1} - c_i)}{T}. \quad (4)$$

Example 1: Let a given time-series be $\vec{C} = [c_1, c_2, c_3, c_4, c_5] = [3,4,6,2,1]$ where the values are sampled at time-instants 1, 2, 3, 4 and 5 respectively. The points in 2-dimensional space formed by the time-series data points as ordinates and the time-instants as abscissas are respectively, $p_1 \equiv (1,3)$, $p_2 \equiv (2,4)$, $p_3 \equiv (3,6)$, $p_4 \equiv (4,2)$ and $p_5 \equiv (5,1)$. Then the edge $l_{1,2}$ formed between the two consecutive data points c_1 and c_2 is given by the straight line equation between points p_1 and p_2, as given in equation (5).

$$l_{1,2} \equiv \frac{y-3}{x-1} = \frac{4-3}{2-1} \Rightarrow x - y + 2 = 0 \quad (5)$$

Clearly, the slope of the edge $l_{1,2}$ is $slope(p_1, p_2) = +1$. In a similar manner, the slopes for the remaining edges $l_{2,3}$, $l_{3,4}$ and $l_{4,5}$ of the time-series are given by $slope(p_2, p_3) = +2$, $slope(p_3, p_4) = -4$ and $slope(p_4, p_5) = -1$ respectively. □

Definition 6: A finite sequence of consecutive edges in a time-series is called an **edge-window**. Let *w* be the chosen width of each edge-window. Then the edge-window starting at the i^{th} time-series data point is given as follows:

$$\vec{W}_i = [l_{i,i+1}, l_{i+1,i+2}, ..., l_{i+w-1,i+w}]. \quad (6)$$

For a given edge-window, we also define a **slope-window** which contains the slope of every edge in the edge-window. For the window given in equation (6), the corresponding slope-window is given as

$$\vec{S}_i = [slope(p_i, p_{i+1}), slope(p_{i+1,} p_{i+2}), ..., slope(p_{i+w-1}, p_{i+w})]. \quad (7)$$

The amount of variation in the slopes contained in a slope window provides insight into the behavior of the time-series for the corresponding temporal region. The higher the variation of slopes, the greater the chances of the time-series displaying a change in its behavioral pattern. Hence, the variation of slopes in a slope-window is used to derive the segmentation criterion of the proposed algorithm.

Example 2: In this example, we demonstrate the formation of edge-windows and slope-windows on the time-series given in Example 1. Let the chosen window width be 3. Hence, the edge-windows formed from the time-series are $\vec{W}_1 = [l_{1,2}, l_{2,3}, l_{3,4}]$ and $\vec{W}_2 = [l_{2,3}, l_{3,4}, l_{4,5}]$. The corresponding slope-windows of the time-series are

$$\vec{S}_1 = [slope(p_1, p_2), slope(p_{2,} p_3), slope(p_3, p_4)] = [+1, +2, -4], \quad (8)$$

$$\vec{S}_2 = [slope(p_2, p_3), slope(p_{3,} p_4), slope(p_4, p_5)] = [+2, -4, -1]. \quad (9)$$

Segmentation criterion: As mentioned before, the occurrence of high variance in the slopes of a slope-window indicates an underlying change in the structural pattern of the time-series at the corresponding temporal region. Hence, in the proposed algorithm, we choose high variance slope-windows to form segment boundaries in the time-series. Let $\sigma(\vec{S})$ denote the variance of the slopes in the slope window $\vec{S}$. If the variance $\sigma(\vec{S})$ is greater than a user-provided threshold T_h, the slope window $\vec{S}$ is chosen to form a segment boundary. Let $l_{i,i+1}$ be the edge occurring in the middle of the chosen slope-window. Clearly, there are two time-series data points c_i and c_{i+1} at each end of the edge $l_{i,i+1}$. In the current paper, we place the segment boundary at c_{i+1} although that is a completely arbitrary decision. Other possible choices for placing the segment boundary can be the point c_i or the mean of c_i and c_{i+1}.

B. Proposed Segmentation Algorithm

The proposed segmentation technique has three main stages. The first stage involves pre-processing of the time-series in order to remove low level noisy fluctuations which might perturb the decision of the segmentation algorithm. There are quite a few ways of doing this. In this paper, we employ the well-known Gaussian smoothing [15] technique in order to smooth the time-series to remove low level noisy fluctuations. The second stage employs the segmentation algorithm to identify segment boundaries in the time-series. However, it may so happen that two consecutive segment boundaries are placed very close to each other if the high variance condition for segmentation persists over multiple consecutive slope-windows. There are a few ways to avoid this scenario. The first is to shift the slope-window by more than one unit in the direction of increasing time, whenever a segment boundary is placed. This will reduce the possibility of two consecutive slope-windows satisfying the high variance segmentation criterion. The second approach is to re-process the segmented time-series such that the multiple closely placed segment boundaries are replaced by a single boundary. In this paper, we adopt the latter option as the third stage in order to maintain coherent shifting of the slope-window over the entire time-series. The three proposed stages of the segmentation technique are illustrated in Figure 3. The EPTS algorithm is formally presented in Pseudo Code 1.

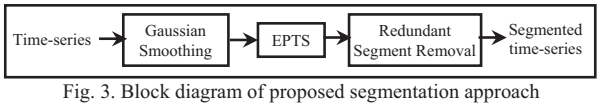

Fig. 3. Block diagram of proposed segmentation approach

Pseudo Code 1: EPTS Algorithm

Input: A time-series of the form $\vec{C} = [c_1, c_2, ..., c_n]$, and user defined parameters, namely the slope-window width w and the slope-variance threshold T_h.

Output: A vector of time-series index values $\vec{I} = [i_1, i_2, ..., i_m]$ denoting the time instants where a segment boundary is placed.

```
BEGIN
  FOR each edge-window W_j in the time-series
  C, j ∈ {1, 2, ..., n−w}, DO
    Compute the slope-window S_j following equation (7);
    Compute the variance σ(S_j) of slope values in the
    window S_j;
    IF σ(S_j) > T_h THEN
      Place a segment boundary at the middle edge of the
      window W_j and at the corresponding time-series data
      point given by c_{j+⌈w/2⌉}. Insert j+⌈w/2⌉ in the vector I;
    ENDIF
  END FOR
END
```

IV. Fuzzy Segment Matching

In this section, we propose a technique to match each temporal segment obtained in the previous step to one of four basic primitive patterns: i) linear rise, ii) linear fall, iii) Gaussian bell curve, iii) inverted Gaussian bell curve as shown in Figure 4. The task of classifying a temporal segment to one of a given set of shapes can be performed in a number of ways. A classic approach is to train a pattern classifier to identify the class or shape to which a temporal segment bears maximum similarity. However, such an approach requires a relatively large dataset to be created for the purpose of training the classifier. Furthermore, the training process also involves considerable computational cost. Hence, we propose a matching scheme which does not involve any training and can work in real time to identify the representative pattern for a given temporal segment.

For the purpose of matching, each segment is first represented by a vector of length 10 so as to achieve uniformity in the temporal length of the segments. Next the segments are mean-normalized in order to remove the effects of varying ranges of different segments which may lead to incorrect match results. Finally, a fuzzy matching scheme is employed to derive a similarity metric between a given temporal segment and a primitive pattern.

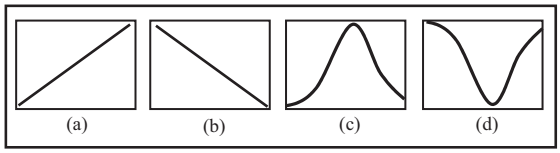

Fig. 4. Four primitive patterns: (a) Linear rise, (b) Linear fall , (c) Gaussian Bell curve, (d) Inverted Gaussian curve

A. Fuzzy Similarity Metric

Let a primitive pattern $\vec{P}$ be represented by a sequence of 10 consecutive points as shown below:

$$\vec{P} = [p_1, p_2, ..., p_{10}]. \quad (10)$$

Let $p_{\max}$ and $p_{\min}$ be the maximum and minimum values of the sequence $\vec{P}$ respectively. We define the range $U = [p_{\min} - k, p_{\max} + k]$ where k is an empirically chosen constant, as the universe of discourse for representing the primitive pattern by a sequence of 10 fuzzy sets, one for each point in the pattern. The fuzzy set defined for the point p_i denotes the linguistic term CLOSE TO p_i. Thus, few of the appropriate membership functions which can be applied to define the fuzzy set are the Gaussian membership function and the triangular membership function, where the maximum value of 1 is attained by the function at the point p_i. In the current paper, we define the triangular membership function for point p_i as follows:

$$\left.\begin{aligned} \mu_i(x) &= \left(\frac{x - p_i + r}{r}\right) \text{ if } (p_i - r) \le x \le p_i \\ &= \left(\frac{p_i + r - x}{r}\right) \text{ if } p_i \le x \le (p_i + r) \\ &= 0 \text{ otherwise} \end{aligned}\right\} \quad (11)$$

where r is defined as a certain percentage of the range $(p_{\max} - p_{\min})$. Given a mean-normalized temporal segment $\vec{S} = [s_1, s_2, ..., s_{10}]$, the similarity between the segment $\vec{S}$ and pattern $\vec{P}$, is defined as the summation of the individual membership values $\mu_i(s_i)$ of each point s_i in the corresponding fuzzy sets for the pattern $\vec{P}$. This is given in equation (12) and is illustrated in Figure 5.

$$similarity(\vec{S}, \vec{P}) = \sum_{i=1}^{10} \mu_i(s_i) \quad (12)$$

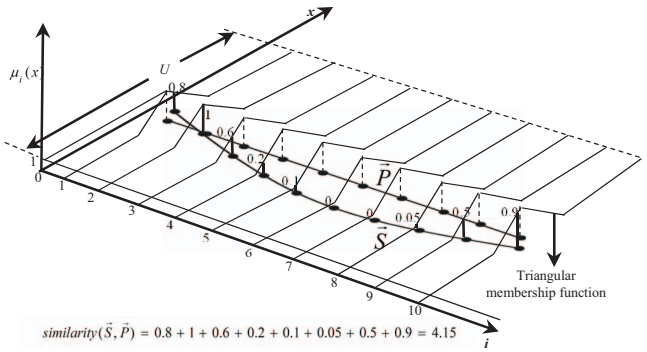

Fig. 5. Computation of fuzzy similarity metric between a given pattern $\vec{P}$ and a normalized temporal segment $\vec{S}$. A triangular fuzzy membership function is used for each of the 10 points of the pattern where U represents the universe of discourse.

B. Fuzzy Matching Algorithm

Having derived the fuzzy similarity metric in the previous step, the matching scheme becomes relatively straightforward. For every normalized temporal segment $\vec{S}$, its fuzzy similarity is computed with each of the four primitive patterns. The segment is classified to pattern $\vec{P}_i$ if

$$similarity(\vec{S},\vec{P}_i) \geq similarity(\vec{S},\vec{P}_j)\ \forall j \in \{1,2,3,4\}. \quad (13)$$

A point worth mentioning here is that there may be certain temporal segments which do not properly represent any of the four primitive patterns which we have used. Hence, we keep a separate class of such outlier (or anomalous) segments. Let a segment $\vec{S}$ be classified to a pattern $\vec{P}_i$ following equation (13). The segment is called an outlier if

$$similarity(\vec{S},\vec{P}_i) < R_t \quad (14)$$

where R_t denotes a user-provided rejection threshold. In essence, if the segment is not similar enough to any of the given patterns, it is an outlier. The fuzzy matching algorithm is formally presented in Pseudo Code 2.

Pseudo Code 2: Fuzzy Matching Algorithm

Input: A normalized segment $\vec{S} = [s_1, s_2, ..., s_{10}]$, a set of four primitive patterns $\boldsymbol{P} = \{\vec{P}_1, \vec{P}_2, \vec{P}_3, \vec{P}_4\}$ where each pattern is represented by a sequence of 10 points and a rejection threshold R_t.

Output: The pattern $\vec{P}_i$ which represents the segment $\vec{S}$.

```
BEGIN
  FOR each pattern P_i in the set P, DO
    Compute fuzzy similarity metric similarity(S, P_i) between
    S and P_i following equation (12);
  END FOR
  IF similarity(S, P_i) ≥ similarity(S, P_j) ∀j AND
  similarity(S, P_i) ≥ R_t THEN
    Output P_i as the representative pattern;
  ENDIF
END
```

V. Predictor Graphs

This section deals with the representation of knowledge obtained from the previous two steps in the form of a set of four weighted directed graphs. The nodes in each graph represent horizontal partitions of the time-series and the weighted directed edge between two nodes represents the transitions from the partition of the head node to the partition of the tail node caused by the occurrence of one or more temporal segments. In the prediction phase, given a certain starting partition, each graph is used to make a prediction on a possible future value of the time-series. The overall forecast of the model is the mean of the individual predictions. The primary advantage of the proposed prediction scheme is that the effects of different primitives are separately learned and utilized to make individual predictions, thereby providing the flexibility of predicting the outcomes of individual primitives on the time-series.

A. Construction of predictor graphs

We define and construct four graphs $\{G_1, G_2, G_3, G_4\}$ by using the following steps:

Step 1. The time-series is partitioned into q equi-spaced intervals or partitions (Definition 4). In each graph $G_p, p \in \{1,2,3,4\}$, a vertex v_i is constructed for every partition p_i. Hence, a total of q vertices are constructed in each graph.

Step 2. Let there be d temporal segments represented by pattern $\vec{P}_x$, which cause a transition from partition p_i to other partitions. If d' of the above mentioned segments produce a transition to partition p_j, we insert a directed arc from the vertex v_i to vertex v_j in the graph G_x (corresponding to $\vec{P}_x$). The weight attached to this newly created arc is given by

$$\delta(v_i, v_j) = \left(\frac{d'}{d}\right) \quad (15)$$

which approximates the probability of transition from partition p_i to partition p_j, due to the occurrence of the pattern $\vec{P}_x$. This step is carried out for every pattern. It should be noted that the sum of the weights attached to edges coming out from a vertex is 1.

Figure 6 illustrates the construction of four predictor graphs from a sample synthetic time-series which has been segmented and classified.

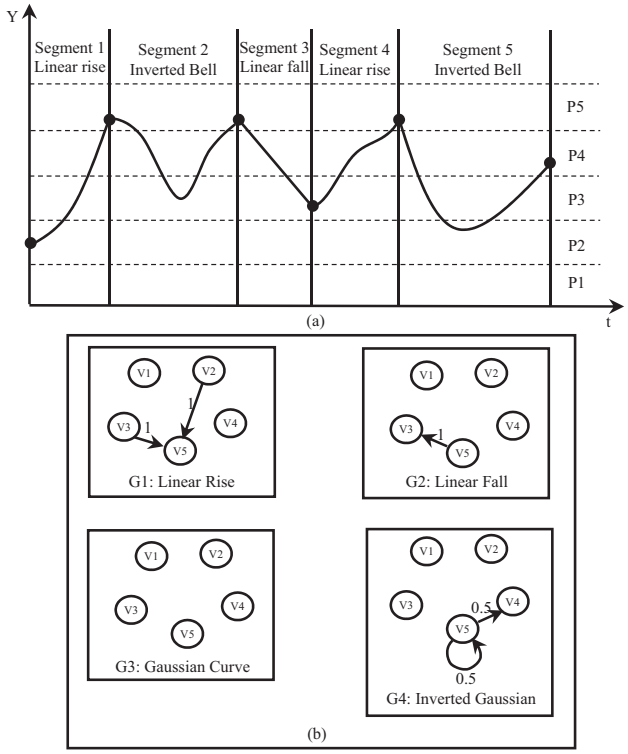

Fig. 6. Construction of predictor graphs from a given segmented, classified and partitioned time-series: (a) A synthetic time-series is segmented and each temporal segment has been classified. For the purpose of constructing predictor graphs, the time-series has been partitioned into 5 partitions. (b) Predictor graphs constructed from the processed time-series shown in (a).

B. Prediction of future time-series values

Once the predictor graphs are constructed, the model can be used to make predictions on possible future values of the time-series. Let the partition corresponding to the current day be p_{cur}. Then, for each graph $G_p, p \in \{1,2,3,4\}$, the prediction is given by

$$Pr_p = \sum_{i=1}^{N} (\delta(v_{cur}, v_i) \times mid(p_i)) \quad (16)$$

where N is the total number of partitions, $\delta(v_{cur}, v_i)$ is the edge-weight of the directed edge from the vertex v_{cur} corresponding to p_{cur} to the vertex v_i, and $mid(p_i)$ represents the mid-point of the partition p_i. In essence, the prediction is a weighted sum of partition mid-points to which a possible transition can take place due to the occurrence of a temporal segment. The overall prediction of the model is given by the mean of the individual predictions as shown below.

$$Pr = \frac{1}{4} \sum_{p=1}^{4} Pr_p \quad (17)$$

VI. Experiments and Results

In this section, we present the results obtained by performing experiments on the TAIEX [14] economic close-price time-series for the period 1990-1999. For each year, we partition the time-series into seven equi-spaced partitions, namely, EXTREMELY LOW, VERY LOW, LOW, MEDIUM, HIGH, VERY HIGH and EXTREMELY HIGH. A higher number of partitions is not preferable as the transitions that we observe are caused by temporal segments consisting of multiple consecutive time-series data points and hence, only the partitions containing the end-points of a segment possess an incoming or outgoing arc in the graph. A large number of partitions will produce many nodes in the predictor graph many of which will never be connected by an edge.

For each year, from 1990 to 1999, the model is trained for the months January to October and the prediction performance is tested on the months November and December. It should be noted that the model predicts the possible time-series data point which will be obtained after the duration of an average segment length. Hence, for each prediction we consider the starting partition of the day earlier from the day of prediction by the average segment length. Figure 7 illustrates the segmented and partitioned time-series as well as the predictor graphs obtained on the training phase for the economic years 1991, 1994, and 1997. Table 1 provides a comparative analysis of the proposed model in comparison with other existing models in the literature. The well-known RMSE error metric is used to test the performance of the models. The results for the other existing models are obtained from [12]. Each of these models are trained on the TAIEX with the training period from January to October and the test period consisting of the months November and December for each year from 1990 to 1999.

The experiments are carried out using MATLAB as a coding platform under a Windows 7 OS powered by Intel Core i7 processor with a system clock of 2.30 GHz and a RAM capacity of 8 GB. The average training time, over the experimental period of 1990-1999 is 2.6832 seconds.

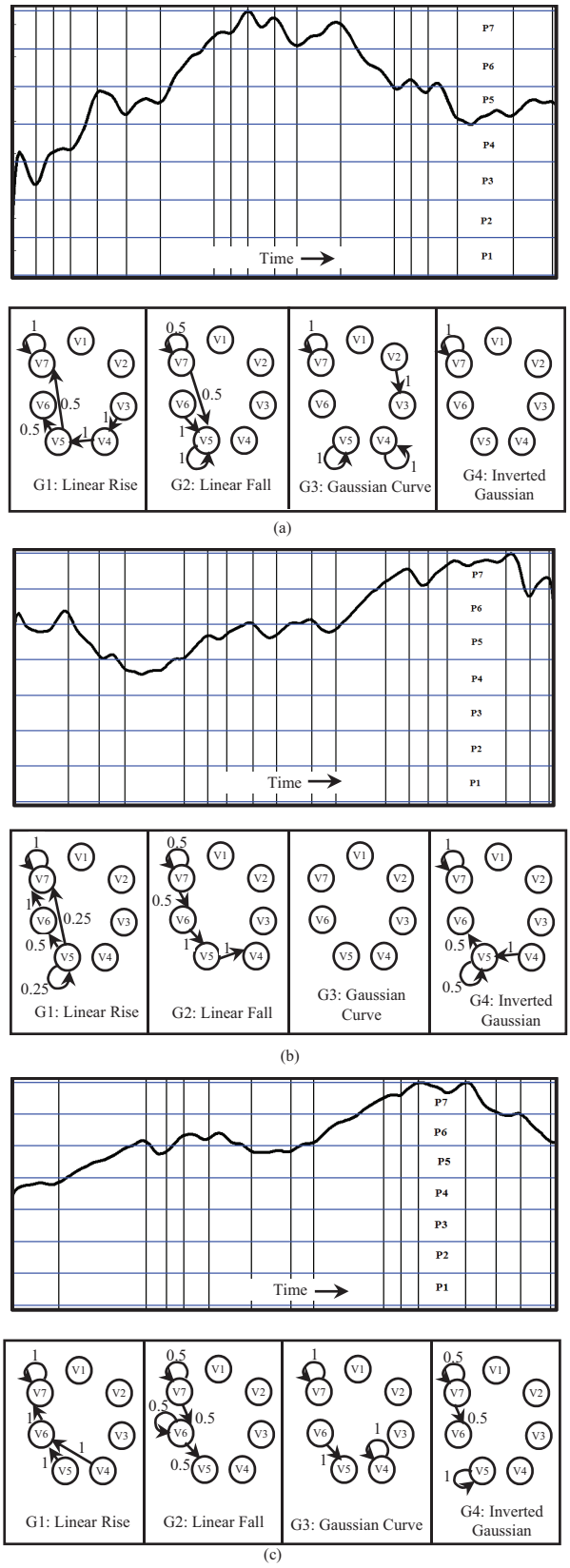

Fig. 7. Predictor graphs obtained from segmented TAIEX close-price time-series for the years (a) 1991, (b) 1994, (c) 1997. The entire TAIEX time-series is partitioned into intervals P1 to P7 (shown by horizontal lines), and the close-price series for each year is separately segmented where the segment boundaries are shown by vertical lines.

TABLE I
COMPARISON OF PREDICTION PERFORMANCE OF PROPOSED MODEL WITH EXISTING MODELS USING THE RMSE METRIC

Models \ Years	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	Average
1.Conventional models [5]	220	80	60	110	112	79	54	148	167	149	117.9
2.Weighted models [5]	227	61	67	105	135	70	54	133	151	142	114.5
3.Chen and Chen [13]	172.89	72.87	43.44	103.21	78.63	66.66	59.75	139.68	124.44	115.47	97.70
4. Chen *et. al* [8]	174.62	43.22	42.66	104.17	94.6	**54.24**	50.5	138.51	117.87	101.33	92.17
5. Chen and Kao [11]	**156.47**	56.50	36.45	126.45	**62.57**	105.52	51.50	125.33	104.12	**87.63**	91.25
6.Cai *et. al* [12]	187.10	39.58	39.37	**101.80**	76.32	56.05	**49.45**	123.98	118.41	102.34	89.44
7.Proposed model	184.65	**34.42**	**35.21**	109.63	67.96	78.21	52.34	**115.81**	**101.69**	104.83	**88.48**

An important point worth noting is that the experimental parameters for each algorithm are chosen empirically. For the EPST segmentation algorithm, we obtain the best results with a slope-window width of 10 data points. The slope variance for all the windows are computed and normalized to lie within the range [0, 1]. The threshold is then chosen as 0.8 for our experiments. For the fuzzy matching algorithm the minimum possible value of fuzzy similarity is 0 and the maximum possible value is 10. A value of 2 for the rejection threshold parameter provides the best results in our experiments.

It is evident from Table 1 that the proposed model, with an average RMSE error of 88.48 outperforms the next best model by 1.0733 % for the prediction of TAIEX close-price time-series. This indicates that the model can be effectively used for real life stock market trading and investment applications.

We have performed a non-parametric statistical test known as the Friedman's test [16] on the RMSE values obtained in Table 1. The number of blocks (years) N for the test is 10 (i.e., the years 1990 – 1999) and the number of treatments (algorithms) k is 7. For each block, the methods are ranked according to the RMSE error values. The average Friedman ranks for the methods over all the years is summarized in Table 2.

TABLE II
AVERAGE RANKINGS OBTAINED THROUGH FRIEDMAN'S TEST

Algorithm	Average Friedman Rank
Conventional models [5]	6.35
Weighted models [5]	5.55
Chen and Chen [13]	4.50
Chen *et. al* [8]	3.10
Chen and Kao [11]	3.10
Cai *et. al* [12]	2.60
Proposed model	2.80

Let the average Friedman rank of the j^{th} algorithm be denoted by R_j. According to the null hypothesis, all the algorithms are equivalent and hence their individual mean Friedman ranks R_j, should be equal. To disprove the null hypothesis we use the Friedman statistic [16] which is computed as follows:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]. \quad (18)$$

For the experiment, this statistic is distributed according to χ_F^2 with k-1 (=6) degrees of freedom. The value of χ_F^2 computed following equation (18) is 28.2756 and the critical value of the same with 7 degrees of freedom for $\alpha = 0.05$ is 12.592. Due to the significant difference between these two values, the null hypothesis is rejected.

We also conduct a post-hoc analysis using the Bonferroni-Dunn's test [17] from the results of the Friedman's test. The control algorithm for this analysis is the one having the lowest mean Friedman rank, i.e., Cai *et. al* [12]. The critical difference value for the Bonferroni-Dunn's test is computed as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}. \quad (19)$$

where the value of q_α for $\alpha = 0.05$ is obtained as 2.394 from [18]. Hence, the critical difference CD is computed as 2.313 following equation (19). The difference between the performances of two algorithms is statistically significant if their individual mean Friedman ranks differ by at least the critical difference CD. The Friedman ranks of all the algorithms are illustrated as a bar chart in Figure 8.

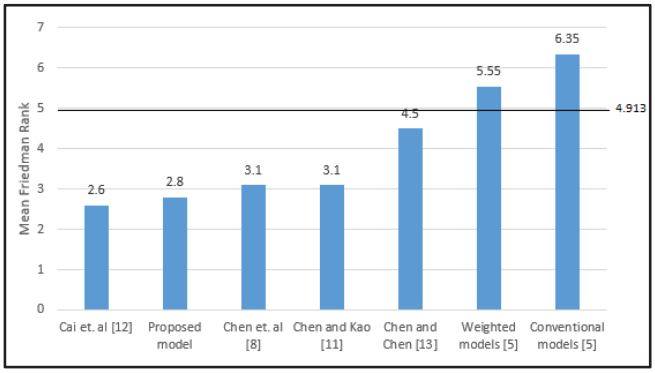

Fig. 8. Graphical representation of Bonferroni-Dunn's test using Cat *et. al* [12] as the control method

It can be inferred from Figure 8 that the difference in the performances of the control algorithm and the proposed model are not statistically significant. Furthermore, the Weighted models [5] and the Conventional models [5] are significantly worse than our proposed model.

VII. CONCLUSION

In this paper, we have presented a novel technique for modeling a time-series with effective applications in forecasting. The proposed model comprises three stages. The first stage is used for semantic segmentation of the time-series

into distinct temporal segments of non-uniform length such that each segment possesses a coherence in its dynamic slope. The proposed EPTS algorithm provides a computationally light-weight approach to finding optimal segment boundaries in the time-series. The second stage employs a fuzzy matching technique to match the temporal segments obtained in the first step with one of four primitive patterns, i) linear rise, ii) linear fall, iii) Gaussian bell curve and iv) inverted Gaussian curve. The proposed fuzzy matching scheme does not require any training and can classify segments in real-time thus, providing an edge over traditional pattern classifiers which have to be trained on a dataset.

The third stage represents the acquired information in the form of a set of four weighted di-graphs corresponding to the four primitive patterns. The graphs can be directly used for prediction or can also be used to generate weighted probabilistic transition rules of first or higher order. Furthermore, the graphs individually provide an insight into the effects of each of the primitive patterns on the time-series. Experiments carried out on the TAIEX close-price economic time-series indicate a low RMSE error value compared to existing fuzzy time-series models in the literature. Thus, the proposed model can be used effectively for time-series forecasting and prediction applications.

REFERENCES

[1] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[2] Q. Song and B.S. Chissom, "Fuzzy Time Series and Its Model," *Elsevier, Fuzzy Sets Syst.*, vol. 54, no. 3, pp. 269–277, 1993.

[3] Q. Song and B.S. Chissom, "Forecasting Enrollments With Fuzzy Time Series—Part I," *Elsevier, Fuzzy Sets Syst.*, vol. 54, no. 1, pp. 1–9, 1993.

[4] Q. Song and B.S. Chissom, "Forecasting Enrollments With Fuzzy Time Series—Part II," *Elsevier, Fuzzy Sets Syst.*, vol. 62, no. 1, pp. 1–8, 1994.

[5] H. K. Yu, "Weighted Fuzzy Time Series Models for TAIEX Forecasting," *Elsevier, Physica A*, vol. 349, no. 3-4, pp. 609-624, Apr. 2005.

[6] K. Huarng, "Heuristic Models of Fuzzy Time Series for Forecasting," *Elsevier, Fuzzy Sets and Systems*, vol. 123, no. 3, pp. 369-386, Nov. 2001.

[7] K. Huarng, T. H. K. Yu and Y. W. Hsu, "A Multivariate Heuristic Model for Fuzzy Time-Series Forecasting," *IEEE Trans. Sys. Man Cybern. Part B,* vol. 37, no. 4, pp. 836-846, Aug 2007.

[8] S.M. Chen, H.P. Chu and T.W. Sheu, "TAIEX Forecasting Using Fuzzy Time Series and Automatically Generated Weights of Multiple Factors," *IEEE Trans. Sys. Man Cybern. Part A,* vol. 42, no. 6, pp. 1485-1495, 2012.

[9] S.T. Li, Y.C. Cheng and S.Y. Lin, "A FCM-Based Deterministic Forecasting Model for Fuzzy Time Series," *Elsevier, Computers and Mathematics with Applications,* vol. 56, no. 12, pp. 3052-3063, Dec. 2008.

[10] S.S. Gangwar and S. Kumar, "Partitions Based Computational Method for High-Order Fuzzy Time Series Forecasting," *Elsevier, Expert Systems with Applications,* vol. 39, no. 15, pp. 12158-12164, Nov. 2012.

[11] S.M. Chen and P.Y. Kao, "TAIEX Forecasting Based on Fuzzy Time Series, Particle Swarm Optimization Techniques and Support Vector Machines," *Elsevier, Information Sciences,* vol. 247, pp. 62-71, Oct. 2013.

[12] Q. Cai, D. Zhang, W. Zheng and S.C.H. Leung, "A New Fuzzy Time Series Forecasting Model Combined with Ant Colony Optimization and Auto-Regression," *Elsevier, Knowledge Based Systems,* vol. 74, pp. 61-68, 2015.

[13] S.M. Chen and C.D. Chen, "TAIEX Forecasting Based on Fuzzy Time Series and Fuzzy Variation Groups," *IEEE Trans. Fuzzy Syst.,* vol. 19, no. 1, pp. 1-12, 2011.

[14] TAIEX [Online]. Available: http://www.twse.com.tw/en/products/indices/tsec/taiex.php

[15] L.G. Shapiro and G.C. Stockman, "Computer Vision," pp. 137-150, Prentice Hall, 2001.

[16] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*,4th ed. Boca Raton, FL, USA: Chapman and Hall/CRC, 2007.

[17] S. Picek, M. Golub, and D. Jakobovic, "Evaluation of crossover operator performance in genetic algorithms with binary representation," in *Proc. 7th Int. Conf. Intell. Comput. Bio-Inspired Comput. Appl.*, 2011, pp. 223–230.

[18] Zar, J.H., Biostatistical Analysis. Prentice Hall. 1999.