CHAPTER 9

FORECAST-ERROR APPROXIMATION BASED BOTTOM-UP SEGMENTATION FOR TIME DELAY NEURAL NETWORK CLASSIFICATION

Clinical time series data contain large set of time-stamped data points that describe the state of a patient's health. Moreover, the observations of these data points are done at irregular intervals and hence knowledge mining from these data becomes challenging. To overcome the complexity in mining such data, there is a need to reduce the dimension (length) of time series data into smaller representations without any loss of information. This research contribution presents a Forecast-Error Approximation based Bottom-Up (FeAB) segmentation approach, for effectively classifying clinical time series data using Time Delay Neural Network (TDNN). The proposed approach includes two functionalities namely temporal data summarization and classification. In temporal data summarization, clinical time series data is divided into a sequence of temporal interpreted segments using the proposed FeAB segmentation. FeAB adopts a Double Exponential Smoothing (DES) technique to derive the growth rate, mean and forecast-error for each clinical observation. The obtained forecast-error is used to compute the merge-cost for FeAB segmentation. TDNN is used in the classification process to build a classification model for the segmented time series.

9.1 SYSTEM FRAMEWORK FOR FeAB SEGMENTATION AND CLASSIFICATION

The framework for the proposed system is shown in the Figure 9.1.

Figure 9.1 System Framework for FeAB Segmentation and Classification

The system comprises of two subsystems, namely temporal summarization and classification. The detailed descriptions of the subsystems and its methods are explained in the forthcoming sections.

9.1.1 Temporal Summarization

The temporal summarization process aims at reducing the size (length) of time series data through effective time series data representations. This is considered as an important data pre-processing technique, which reduces the complexity of temporal data classification. Temporal summarization performs the following activities: temporal smoothing and segmentation. Temporal smoothing transforms the quantitative values of each attribute into qualitative temporal interpretations such as trend and state. Trend represents the growth rate and state describes mean value of the clinical attributes. The obtained trend and state for each clinical observation has been used in computing its forecasted value. The process of segmentation divides the sequence of long time series into short subsequences, which are referred to as segments. The segments are then summarized using the obtained temporal interpretations.

9.1.1.1 Temporal Smoothing: Hanzak Improved DES

Temporal smoothing computes the temporal interpretations like growth rate, mean and forecast value for each clinical observation. This work adopts an improved DES method presented by Cipra & Hanzak (2008) to perform the smoothing process. The traditional DES method uses two smoothing constant parameters α and β to estimate the growth rate and mean value respectively (Holt 2004; Holt 1957). The parameter α and β can take any value in the range of 0 and 1 and this value remains constant in all the estimations. The wrong choice of value for α and β will affect the accuracy of forecasted results. Also, the traditional DES assumes time series to be regular

with even interval spacing's. To overcome these limitations of the DES, Wright (1986) suggested that the value for smoothing constant parameter can be adjusted based on the interval spacing's among the observations. Time-close observations in time series data arises when the interval between two observations are too small. Cipra & Hanzak (2008) addressed this problem of time-close observations in DES method for supporting irregular time series data analysis. This research contribution has adopted Hanzak updated DES to perform temporal smoothing.

The research contributions presented in the chapters five and eight have also adopted DES methods. However, the contribution presented in the current chapter differs from earlier contributions in the following ways: First, in the choice of adopted DES method and Second, in the usage of forecasted results. The contribution presented in the fifth chapter has used Wright updated DES and the contribution presented in the eighth chapter has proposed a FIDES method that incorporates fuzzy logic in DES to compute the trend, mean and forecasted value for each clinical observation. Also, the contributions presented in chapters five and eight have used forecasted value to impute missing values in time series data, whereas the contribution presented in the current chapter has used forecasted value to compute the merge cost in the segmentation process.

The computations done using Hanzak enhanced DES techniques are as follows: smoothing constant (α and β) updation, mean estimation, growth rate estimation and forecast value estimation. Let $Y_{t_n} = \{Y_{t_n}(i),\ t_n \in Ts, i \in A\ ; t_{n+1} > t_n\}$ be an unevenly spaced time series, t_n is the observation time point for a patient, Ts is the set of observation time points, A is the attribute set that represents set of all clinical laboratory test undergone by a patient, $P_{t_{n+1}}(i)$ is the mean value for ith clinical lab test at time t_{n+1}, $Q_{t_{n+1}}(i)$ is the growth rate for ith clinical lab test at time t_{n+1}, $\alpha_{t_{n+1}}(i)$ is the smoothing constant parameter for the mean, $\beta_{t_{n+1}}(i)$ is the smoothing

constant parameter for the growth rate of i[th] clinical laboratory test at time t_{n+1}, $E_{t_n}(\mathrm{i})$ is the error rate for i[th] clinical lab test at time t_{n+1} and $F_{t_{n+2}}(\mathrm{i})$ is the forecasted value for i[th] clinical lab test at time t_{n+2}. $\alpha_{t_0}(\mathrm{i})$ and $\beta_{t_0}(\mathrm{i})$ is the initial smoothing constant parameter for the mean and growth rate of i[th] clinical laboratory test. The smoothing constants for each clinical observation are updated based on the interval days between each observation as defined in the equation (9.1), (9.2), (9.3) and (9.4).

$$\alpha_{t_0}(i) = 1 - (1 - \alpha)^{\left(\frac{q}{\delta_{max}}\right)} \tag{9.1}$$

$$\beta_{t_0}(i) = 1 - (1 - \beta)^{\left(\frac{q}{\delta_{max}}\right)} \tag{9.2}$$

Where q and δ_{max} is average and maximum interval spacing for the laboratory test.

$$\alpha_{t_{n+1}}(i) = \alpha_{t_n}(i)/(\alpha_{t_n}(i) + (1 - \alpha)^{(t_{n+1} - t_n)}) \tag{9.3}$$

$$\beta_{t_{n+1}}(i) = \beta_{t_n}(i)/(\beta_{t_n}(i) + \frac{t_n - t_{n-1}}{t_{n+1} - t_n}(1 - \beta)^{(t_{n+1} - t_n)}) \tag{9.4}$$

The mean estimate uses $\alpha_{t_{n+1}}(i)$ to calculate the mean value of an observed clinical attribute at a specified time say $\mathrm{t_{n+1}}$. This is defined in the Equation (9.5).

$$P_{t_{n+1}}(i) = \alpha_{t_{n+1}}(i) Y_{t_{n+1}}(i) + \left(1 - \alpha_{t_{n+1}}(i)\right)\left(P_{t_n}(i) + (t_{n+1} - t_n) Q_{t_n}(i)\right) \tag{9.5}$$

Growth rate estimate uses $\beta_{t_{n+1}}(i)$ to compute the growth rate value of an observed clinical attribute at a specified time say $\mathrm{t_{n+1}}$. Equation (9.6) defines the growth rate estimate.

$$Q_{t_{n+1}}(i) = \beta_{t_{n+1}}(i)\left(P_{t_{n+1}}(i) - P_{t_n}(i)\right)/(t_{n+1} - t_n\ ) + (1 - \beta_{t_{n+1}})Q_{t_n}(i) \quad (9.6)$$

The forecasted value of an observed clinical attribute at a specified time is calculated from its previous observed mean and growth rate using Equation (9.7). The forecast error is computed using Equation (9.8).

$$F_{t_{n+2}}(\mathrm{i}) = P_{t_{n+1}}(\mathrm{i}) + Q_{t_{n+1}}(i) \quad (9.7)$$

$$E_{t_n}(\mathrm{i}) = Y_{t_n}(\mathrm{i}) - F_{t_n}(i) \quad (9.8)$$

The steps followed in the smoothing process to derive the growth rate, mean and forecast value are described below.

Step 1: Extract the clinical time series ($Y_{t_n}(i)$) of size 'n' for a clinical laboratory test (i), where t_n is the observation time point for a patient.

Step 2: Initialize the mean and growth rate parameter α_{t_0} and β_{t_0} using equation (9.1) and (9.2), where $\alpha_{t_0}(i)$ and $\beta_{t_0}(i)$ are the initial estimate of smoothing constants.

Step 3: Get the initial estimates for mean ($P_{t_0}(i)$) and growth rate ($Q_{t_0}(i)$) by using least square estimation.

Step 4: Compute the forecast value ($F_{t_1}(\mathrm{i})$) using P_{t_0} and Q_{t_0}.

Step 5: While (there exist observations for lab test (i)) Repeat steps 6, 7 and 8.

Step 6: Compute the smoothing constant for level α_{t_n} and trend β_{t_n} using equation (9.3) and (9.4).

Step 7: Compute the mean $P_{t_{n+1}}(i)$, growth rate $Q_{t_{n+1}}(i)$ and forecasted value $F_{t_{n+2}}(\mathrm{i})$ using equation (9.5), (9.6) and (9.7).

Step 8: Calculate the forecast error using equation (9.8).

The temporal interpretations for each clinical attribute are obtained from the growth rate estimate ($Q_{t_{n+1}}$) and mean estimate ($P_{t_{n+1}}$) in temporal smoothing process.

9.1.1.2 Forecast-Error Approximation based Bottom-Up (FeAB) segmentation

Segmentation is the process of dividing or representing a lengthier time series into smaller sequences of segments. This is considered as a pre-processing step in many time series analysis (Lovric et al. 2014; Batal et al. 2013; Lin et al. 2007). In clinical time series data, uneven spacing arises due to the irregular observation of patient health status. These uneven spacing challenges the segmentation process. The proposed FeAB handles these uneven spacing and performs an effective segmentation. Let Y be a time series dataset denoted as, $Y = \{Y_{t_n}(i),\ t_n \in Ts, i \in A\ ; t_{n+1} > t_n\ \}$; where t_n is the observation time for a patient, Ts is the set of observation time points and A be attribute set, Y_{t_n}(i) be the time series for the ith clinical laboratory test, which contains n observations. In the segmentation process, the time series Y of length 'n' is represented in 'k' segments. Let Sg denotes the set of segments represented as Sg= {$sg_{b,e}$(ws)}, 1<=ws<=k; $b, e \in Ts$; b<=e}, where b and e represents the boundaries of the segments. Each segment '$sg_{b,e}$(w)' in 'Sg' has a set of consecutive observations, $sg_{b,e}(ws)=\{y_{t_r}(ws); b \le t_r \le e\}$, where t_r represents the observation time points within the appropriate boundaries b and e, $y_{t_r}(ws)$ represents the value measured at t_r in the wsth segment.

Figure 9.2 illustrates an example for a bottom-up segmentation strategy. If there are 'n' observations, the bottom-up strategy initially creates 'n-1' segments at the first level. At the next level it computes the merge cost for each segment and tries to merge the segments, which minimizes the segment error (Lovric et al. 2014; Keogh et al. 1993). Figure 9.2 shows a sample time series that has nine observations and depicts the process of transforming and representing the nine observations into five segments. At the first level (Level-1), eight segments ($sg_{1,3}^{(1)}$, $sg_{3,6}^{(2)}$, $sg_{3,7}^{(3)}$, $sg_{4,10}^{(4)}$, $sg_{5,13}^{(5)}$, $sg_{6,16}^{(6)}$, $sg_{7,19}^{(7)}$, and $sg_{8,23}^{(8)}$) were formed and at the next subsequent levels segments get merged based on the computed merge cost. FeAB adopts this Bottom-up segmentation strategy presented in the literatures (Lovric et al. 2014; Keogh et al. 2004). However, FeAB differs from these literatures in the process of merging the adjacent segments.

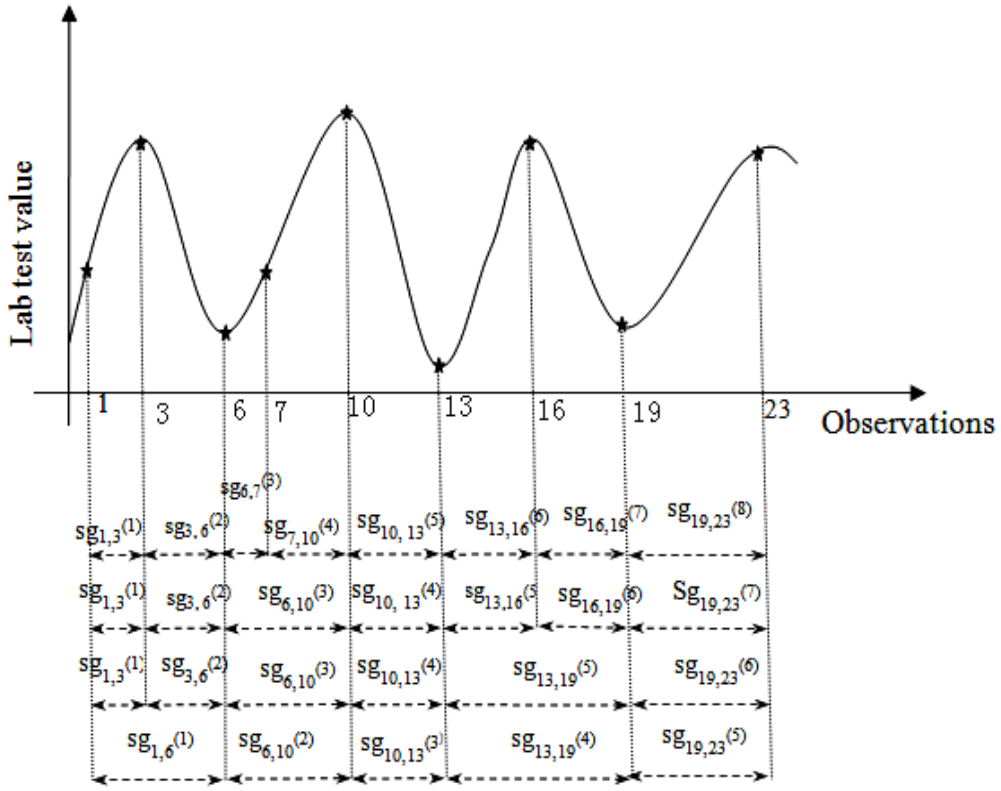

Figure 9.2 Illustration of Segmentation Process

FeAB computes the merge cost error for each segment based on the forecast error approximation value obtained using the Equation (9.9).

$$\text{fRMSE}_{\text{e}}(ws) = \frac{1}{|ns|}\sum_{r=1}^{e}(y_{t_r}(ws) - F_{t_r}(ws)) \; ; \; sg_{t_r}(ws) \in sg_{b,e}; t_r \in ns \quad (9.9)$$

Where e is the boundary of the wsth segment, $y_{t_r}(ws)$ represents the value measured at the segmented point, $F_{t_r}(ws)$ is the forecasted value obtained for the segmented point, t_r represents the observation time points, ns represents the set of observations in wsth segment and $|ns|$ represents the number of observations in wsth segment.

The segments with minimum merge cost get merged in the bottom-up way. The steps followed in the FeAB technique for segmenting clinical time series data are summarized as follows,

Step 1: Obtain the time series Y_{t_n}(i) of size 'n' for ith clinical laboratory test from Y_{t_n}.

Step 2: Create the initial (n-1) segments (Sg) from the time series Y_{t_n}(i) by concatenating subsequent observations.

Step 3: Compute the cost of merging each consecutive segment pairs using the forecasting approximation error. The forecast-error approximation $\text{fRMSE}_{\text{m}}(j)$ is computed for the initial segments using equation (9.9).

Step 4: Identify the segment pair with minimum merging cost.

Step 5: Merge the two segments, update the boundaries and create new segments.

Step 6: Compute the number of segments.

Step 7: While (the computed number of segments (K) is greater than maximum number of segments (Max_Sg)). Repeat Steps 8, 9 and 10.

Step 8: Compute the forecast-error approximation with the updated new segment using equation (9.9).

Step 9: Identify the segment pair with minimum merging cost.

Step 10: Merge the two segments, update the boundaries and create new segments.

Step 11: Aggregate the trend and level obtained for observed time points in each segments.

Step 12: Return the segments and its temporal interpretations.

The growth rate and mean value obtained for each observation in a segment is aggregated and represented as trend and state respectively. The obtained trend is normalized in the range scale of -1 to 1 and the state is normalized in the range scale of 0 to 1 using min-max normalization (Han & Kamber 2001).

9.1.2 Temporal Classification: Time Delay Neural Network (TDNN)

Time Delay Neural Network (TDNN) is a type of feed-forward neural network with tapped delay lines for the inputs. TDNN was introduced by Waibel (1989) for identifying the temporal relationships among the acoustic-phonetic features. TDNN unfolds the inputs in different time points and interprets the temporal sequence. In this work, TDNN is used to build a classification model for clinical time series data. Figure 9.3 shows the network structure of TDNN used in the proposed framework.

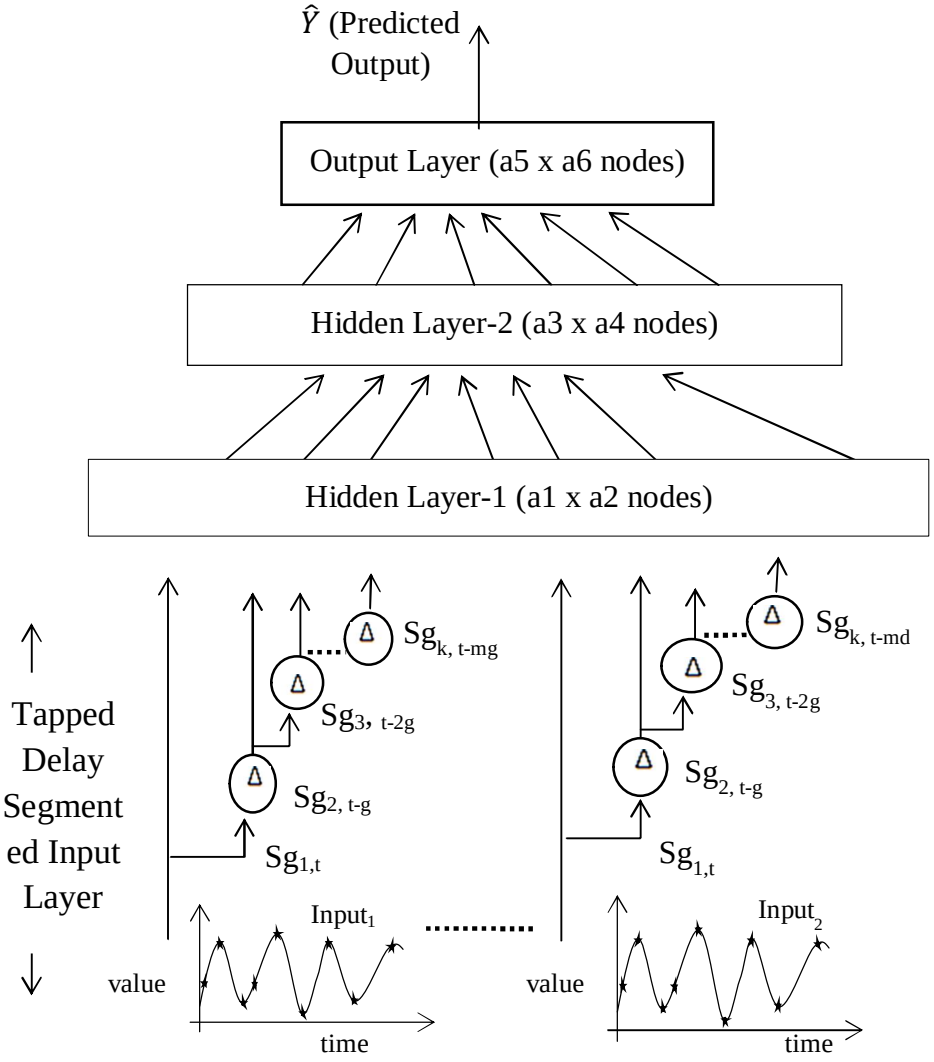

Figure 9.3 Time Delay Neural Network Structure for Classification

The network structure uses two hidden layers. The nodes in the input layer correspond to the attributes in the clinical data. The parameters a1, a2, a3, a4, a5 and a6 given in the Figure 9.3 define the dimensions of two hidden layers and output layer. The first hidden layer consists of nodes arranged in a1xa2 vector. The second layer consists of nodes arranged in a3xa4 vector. The nodes in the output layer are represented in a5xa6 vector. The attribute in clinical time series data refers to the laboratory test undergone by the patients. For each attribute, its segments and their interpretations are taken as tapped delay inputs to the TDNN.

The training algorithms such as Levenberg-Marquardt (LM), Gradient Descent (GD), Scaled Conjugate Gradient (SCG) and Gradient Descent with Momentum (GDM) learning (Waibel 1989; Hagan & Menhaj 1999; Moller 1993; Hagan et al. 1996) available for feed-forward neural networks can be used to train the TDNN. However, for training the network this work uses the traditional Backpropagation (BP) (Han & Kamber 2001; Hagan & Menhaj 1999) with LM learning technique. In this technique, the network's weights are adjusted by back-propagating the error to minimize the classification error. The algorithm performs two major functionalities namely forward input propagation with segmented delay and backward error propagation with weight updation.

In forward input propagation, the segmented inputs are fed forward to the TDNN with the segmented delay of ‘g’. The procedure for training TDNN using back-propagation LM algorithm is discussed in the steps 1 to 7. The forward input propagation with delay includes the steps involved in forwarding the temporal segmented inputs in the forward direction, which is illustrated in the steps 3 to 4. The steps 5 and 6 describe the process of computing the error, backpropagation and weight updation.

The following are the steps used in backpropagation algorithm for training the TDNN:

Step 1: Initialize the weights (w_{ij}) of the network randomly, where i refer to the source node and j refers to the destination node.

Step 2: Read the temporal input sequence of each attribute in TDNN with a segmented delay of g.

// Forward input propagation with delay

Step 3: For every node in the input layer, its output ($O_j(t)$) equals its input ($I_j(t)$) along with the segmented time delay lines given to it as defined in the Equation (9.10).

$$O_j(t) = I_j(t) \tag{9.10}$$

where $I_j(t)$ refers to the node in input layer and ($O_j(t)$) refers to its output, where j (1<=j<=m) refers to the node in the input layer, t refers to the segmented time point and m refers to the number of nodes in input.

Step 4: For every node (j) in the hidden layer, compute its output using the equation (9.11).

$$O_j(t) = \varphi\left(\sum_{i=1}^{m}\sum_{l=0}^{g} w_{ij}x_i(t-l)\right) + \theta_j \tag{9.11}$$

where w_{ij} is the networks weight adjusted through backpropagation, $x_i(t-l)$ inputs at the segmented time $(t-l)$ for ith node in the input layer, m refers to the number of nodes in the input layer, g refers to the delay lag for each input sensor observation, l takes any value from 0 to g. θ_j is the bias and φ denotes the activation.

Step 5: For every node (k) in the output layer, compute its output ($O_k(t)$) using the equation (9.12).

$$O_k(t) = \varphi\left(\sum_{j=1}^{h} w_{ik}O_j(t)\right) + \theta_k \tag{9.12}$$

where w_{ik} is the networks weight adjusted through back-propagation, $O_j(t)$ output from the jth hidden node, h refers to the number of hidden nodes, θ_k is the bias and φ denotes the activation.

// Error Backpropagation and Weight Updation

Step 6: For each node in the output layer, the error (Err_{ij}) is computed based on the predicted and actual value. The weights are updated using the equation (9.13).

$$w_{ij} = w_{ij} + Err_{ij} \tag{9.13}$$

Step 7: The network training stops when the user- specified number of iterations ends.

9.2 EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed temporal summarization and classification process is experimented with the Hepatitis and Thrombosis datasets. These data are randomly divided into two sets, train and test, which contain 80% and 20% of patients respectively. The temporal summarization process starts with extracting the growth rate, mean value and forecasting error using improved DES (Cipra & Hanzak 2008). A dynamic adjustment of the smoothing constant in the growth rate and mean value estimation is done to handle the data irregularities. There are two smoothing factor namely α and β used in mean and growth rate computations respectively. In general, the value of α and β is any value chosen between 0 and 1. When the value of α is closer to 1 it represents that more weight is given to the recent observations. The parameter α is assigned with a small value, when there is a need for stable predictions with smoothed random variation. Whenever a rapid response to a real change in the pattern of observations is needed, a large value of α is appropriate. Similarly, when β is closer to 1 the trend is updated with respect to forecast error. Whenever the value of β is closer to 0, then there is a constant updation in the trend estimate.

This work initializes the smoothing constant α to be 0.4 and β to be 0.7. This initial value for α and β is updated dynamically for each clinical observation based on the interval spacing. The performance measures such as Mean Squared Error (MSE), Mean Absolute Deviation (MAD), error rate, Mean Absolute Percentage Error (MAPE) are used to compare the results of enhanced Holt and classical Holt method (Holt 2004; Cooray 2008). The performances of the forecasting results are evaluated using 10-fold cross validation. Table 9.1 shows the level, trend and forecasted value computed using improved DES for a subset of data taken from Hepatitis patients with Medical Identity (MID) 1, for lab test CRE, in the year 2000. This subset totally had 181 observations. The initial estimates for growth rate $(Q_{t_0})$and mean value $(P_{t_0})$is computed using least square trend estimation.

In the Table 9.1, positive value in the column named trend indicates an upward growth and the negative value indicates a downward growth with respect to the clinical observation done for the patient. It can be inferred from the Table 9.1 that the smoothing constant parameter for ‘α’ and 'β' is updated dynamically based on the interval spacing between the observation. This dynamic adjustment overcomes the complexity of forecasting in unevenly spaced time series data. A statistical paired t-test (Zimmerman & Donald 1997) with significant level of 0.05 is carried out to evaluate the significance of Hanzak updated DES over Wright updated DES. The ρ-value obtained for Hanzak updated DES was found to be less than 0.05. Hence, the forecasting accuracy of Hanzak updated DES was considered to have a significant improvement over Wright updated DES.

Table 9.1 Forecasting Results using Hanzak updated DES for Hepatitis patients [medical identity (MID) 1, lab test-CRE, year- 2000]

Patient (MID)	Observation Date	Value Y_{tn}	Alpha	Beta	Mean P_t	Growth Rate Q_t	Forecast F_t	Error E_t	% Error	RMSE	MAD_t	$MAPE_t$
		-----	0.20	0.30	1.10	-0.400	-----	------	------		------	------
1	19/1/2000	0.78	0.17	0.33	1.04	-0.288	0.70	-0.08	10.26	0.08	0.08	10.26
1	16/2/2000	0.78	0.17	0.95	0.75	-0.022	0.75	-0.03	3.95	0.06	0.06	7.11
1	15/3/2000	0.84	0.17	0.53	0.75	-0.010	0.73	-0.11	12.77	0.08	0.07	8.99
1	19/4/2000	0.80	0.18	0.53	0.75	-0.005	0.74	-0.06	6.92	0.07	0.07	8.48
1	31/5/2000	0.81	0.18	0.47	0.76	-0.002	0.75	-0.06	7.46	0.07	0.07	8.27
1	5/7/2000	0.88	0.18	0.35	0.78	-0.001	0.76	-0.12	13.86	0.08	0.08	9.20
1	30/08/2000	0.80	0.18	0.43	0.78	-0.001	0.78	-0.02	2.66	0.08	0.07	8.27
1	27/09/2000	0.79	0.18	0.34	0.78	0.000	0.78	-0.01	1.04	0.07	0.06	7.36

The second step in the process of temporal summarization is to segment the time series. The FeAB segmentation process represents the time series as segments using a bottom-up strategy that merges each segment pair based on forecasting error. For each segment of a clinical attribute the temporal interpretations are aggregated. This work chooses the number of segments based on the efficiency of segmentation in terms of approximation error observed after conducting several trail experiments

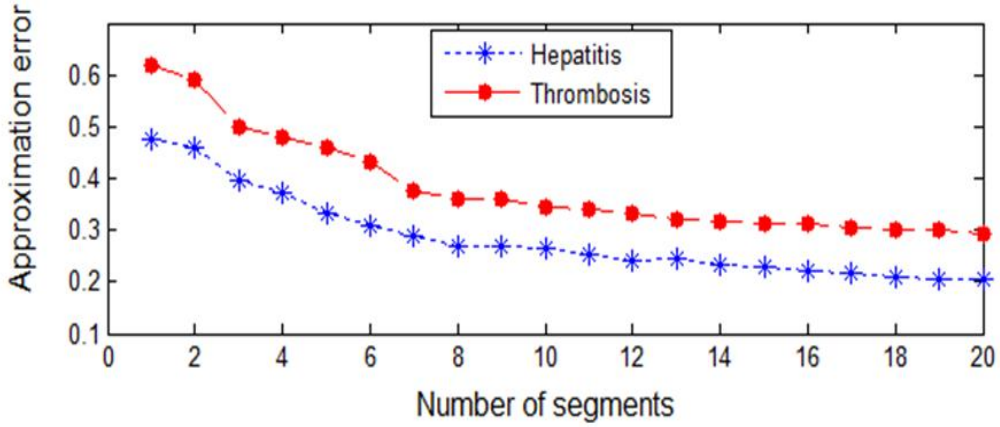

Figure 9.4 Plot for Segmentation Approximation Error Vs Number of Segments

Figure 9.4 shows the graphical plot that illustrates the approximation error obtained for varied number of segments. From the Figure 9.4 it is inferred that as the number of segments increases the approximation error decreases. The approximation error for Hepatitis and Thrombosis dataset is found to decrease when the number of segments is chosen to be more than eight and ten respectively. From the results of the trail experiments and its inference this work chooses the number of segments for Hepatitis dataset as ten and for Thrombosis dataset as eleven. Thus, this work divides the time series into a sequence of ten segments for Hepatitis dataset and eleven segments for Thrombosis dataset.

The growth rate and mean value obtained in each segment are aggregated by taking its average value and represented as trend and state respectively. A TDNN classifies the segmented time series. The network structure of TDNN used in the proposed work consists of one input layer with tapped delay units, two hidden layers and an output layer. The summarized segmented sequence of inputs obtained for each clinical observation is fed to the input layer at different segmented time points. The network parameters such as the number of hidden layers, number of nodes, learning rate and training iterations were selected after conducting various experimental trails. This work uses two hidden layer TDDN. For Hepatitis dataset, the segmented tapped input layer consists of nodes represented in 29x10 vectors.

The first hidden layer consists of 60 nodes arranged in 12x5 vectors. The second hidden layer consists of 18 nodes arranged in 6x3 vectors. The output layer contains two nodes represented in 1x2 vectors. For Thrombosis dataset, the segmented tapped input layer consists of nodes represented in 33x12 vectors. The first hidden layer consists of 90 nodes arranged in 15x6 vectors. The second hidden layer consists of 24 nodes arranged in 8x3 vectors. The output layer contains two nodes represented in 1x2 vectors. The learning rate is chosen as 0.8 and the training iterations is taken as 600. This work uses a Levenberg Marquadt (LM) algorithm for network training. The classification results obtained training algorithms such as LM, SCJ, GDM and GD is shown in the Table 9.2.

Table 9.2 Classification Results of TDNN after FeAB Segmentation

Dataset	TDNN Learning Techniques	Performance Measures								
		Accuracy (%)	Sensitivity (%)	Specificity (%)	Error Rate (%)	Precision	PLR	NLR	PPV	NPV
Hepatitis	LM	91.98	92.03	91.92	8.02	94.54	11.39	0.09	0.95	0.88
	SCJ	85.77	86.78	84.31	14.23	87.37	5.11	0.13	0.87	0.82
	GDM	85.57	85.76	85.29	14.43	86.25	4.64	0.15	0.86	0.81
	GD	81.36	82.37	79.90	18.64	82.65	3.61	0.19	0.83	0.76
Thrombosis	LM	91.17	93.33	88.57	8.83	93.64	8.85	0.09	0.94	0.92
	SCJ	84.42	87.38	80.86	15.58	90.47	5.86	0.12	0.90	0.84
	GDM	85.32	89.76	80.00	14.68	88.77	4.78	0.16	0.89	0.87
	GD	80.52	82.14	78.57	19.48	85.86	3.63	0.20	0.86	0.79

For training TDNN, the best-fit parameter values are selected by performing several trail experiments. Hence, the computational cost associated with training this network is higher compared to other traditional classification approaches. However, this cost is allied only with the training process. The improved classification accuracy using the obtained trained classification model makes it recommended for supporting clinical decision making activities. The trained model can be used for assisting the clinical decision making activities.

9.3 CONCLUSION

This contribution presents a temporal framework that aims at building a classification model for unevenly spaced clinical data. The presented framework consists of two functionalities namely temporal summarization and classification. In temporal summarization the time series data are subjected to the process of smoothing and segmentation. The smoothing process adopts an improved DES (Cipra & Hanzak 2008) to derive the growth rate, mean value and forecasted value. Based on this forecasting error rate the time series are segmented with a bottom-up segmentation strategy. For each segment the temporal interpretations like the trend and state are aggregated. The segments obtained for each clinical attribute are given as tapped delay input to the TDNN for constructing a trained classification model. This contribution uses LM algorithm for training the feed-forward TDNN. There are many challenging areas of research in handling irregularities in the time stamped data and to effectively extract useful knowledge hidden in them.