Chapter 4

Early classification on MTS

The previous chapter addresses the problem of the early classification of UTS. Two methods have been presented by defining two different decision strategies. In this chapter, we address the problem of early classification on MTS by learning optimal decision criteria.

4.1 Introduction

With the advancement in the technology, more and more MTS data have appeared, including an audio stream, video sequences, and sensory information. Thus, numerous applications have been benefited from time series-based data-driven approaches under the purview of data mining and machine learning [108] [109] [110].

In previous studies, early classification approaches maximally developed for UTS [64, 9, 66, 111, 12], that are not applicable directly for MTS. Moreover, an early classification on MTS is a challenging task as compared to UTS because of the presence of multiple variables (for instance, each variable in MTS represents a UTS). Often, these variables are of different lengths and have hidden interconnected relationships. In this regard, a very few notable works have been accomplished using shapelet-based methods [68] [59] [70] [62]. Ghalwash *et al.* proposed a multivariate shapelet detection

(MSD) method and applied it on gene expression data. They extracted multivariate shapelets by employing a sliding window on MTS and selected the key shapelets for early classification using weighted information gain.

The shapelet-based method utilize the local patterns as interpretable features that are extracted from MTS in the training dataset. Thus, irrespective of its interpretability, these methods demand intensive computation [70]. Moreover, the existing approaches for early classification on MTS have not adequately addressed the problem of trade-off optimization. Hence, in this work, we propose an optimization-based early classification approach for MTS data to address the challenges mentioned above.

4.2 Motivation and significant contributions:

The motivations behind this work include: (i) early classification on MTS has many real-world promising applications besides it is being a challenge due to its variability and complexity, (ii) Existing early classification approaches for MTS do not optimize between accuracy and earliness simultaneously while learning decision rules. Hence, this chapter proposes an optimization-based early classification model for MTS data by extending the optimization-based early classification approach presented in the previous chapter, defined for UTS. The novelty of this work lies in the way the proposed model uses ensemble-based classification on MTS data and defines the ESRs to provide a reliable class prediction based on probabilistic outputs of underlying classifiers. The significant contributions of this work are as follows:

- First time, the problem of early classification on MTS has been addressed from an optimization point of view.
- The proposed early classification model for MTS captures temporal information from each variable separately and uses them collectively to make an optimal decision on incoming MTS.
- The proposed model follows a two-layered system in which the first layer defines

a set of PCs and the second layer defines the decision rules.

- First layer introduces the majority voting scheme with tie resolution to predict the class label of incomplete MTS. The second layer defines ESRs for making reliable decisions on incoming MTS and learns its parameters through optimization by considering the misclassification cost and delaying decision cost in its cost function.

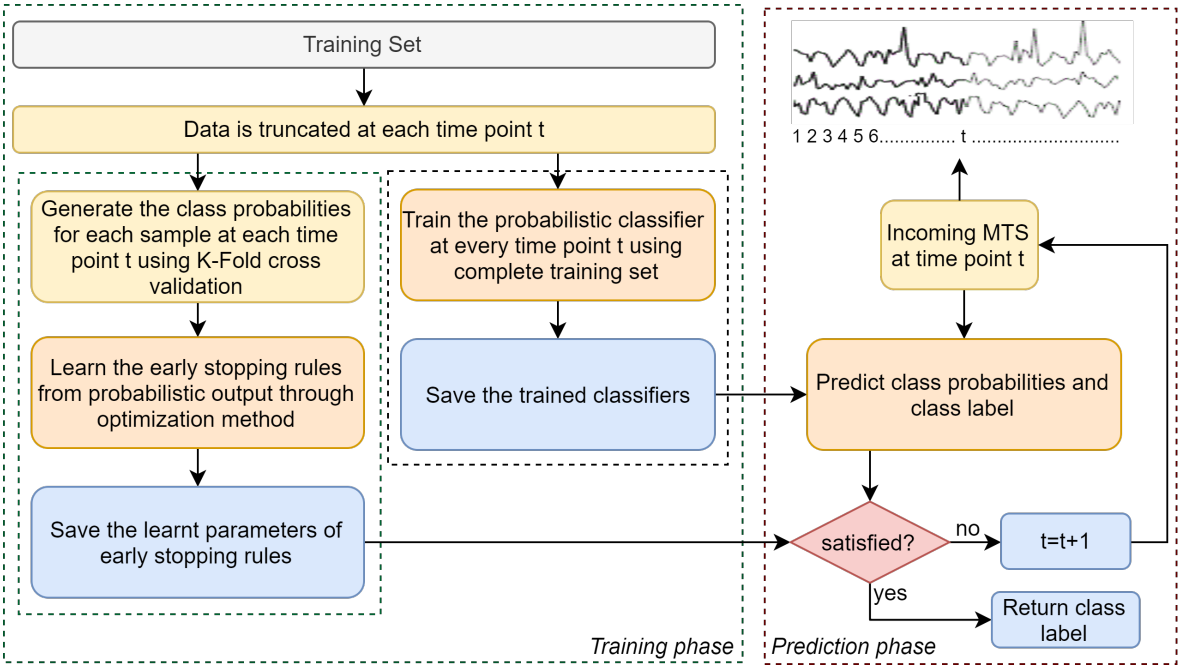


Figure 4.1: Block diagram of the proposed model for early classification on MTS

4.3 Model description

This section provides a complete description of the proposed early classification model for MTS. Figure 4.1 depicts the model in two phases: the training phase and the prediction phase. In the training phase, the model performs the two tasks (i) learning the optimized ESRs from training data, (ii) training the PCs at all defined time points with the full training set. For determining the ESRs, a K-fold cross-validation process is adopted. In each iteration, ad-hoc PCs are trained using (K-1) folds training data and

generate the class probabilities for the other remaining fold (e.g., the fold that is not used for training the ad-hoc classifiers). In this way, class probabilities for a complete training set are generated at all defined time points. Moreover, this probabilistic output of each variable in MTS is utilized to learn the ESRs through the optimization process. In the prediction phase, incoming MTS at each time point t is presented to corresponding PCs, which return the probabilistic output. Moreover, this output is analyzed by the ESR to take the final decision regarding whether to predict the class label or to wait for more data points in the MTS.

4.3.1 Training phase

The objective of this phase is to train the early classification model for MTS using training set $\mathcal{D} = \{(\mathbf{X}^i, y^i), 1 \leq i \leq M\}$ where y is the class label of corresponding MTS $\mathbf{X}$, and M is the number of samples in the dataset. The proposed phase is divided into *four* steps. The *first* step demonstrates the learning process of a series of PCs, which provides the probabilistic output for MTS, and the *second* step defines the ESRs which helps in decision making for early classification. The *third* step defines the cost function that considers the delaying decision cost and misclassification cost in order to optimize the accuracy as well as the earliness. Moreover, this cost function is used to learn ESRs. Finally, in the *fourth* step, the learning procedure of ESR is presented.

***Step 1**: Classifier training*:

A set of PCs $\mathcal{H}_t = \{\{h_t^v\}, 1 \leq v \leq V\}$ are trained at every time point t or explicitly defined by the user based on the knowledge of application domain. Figure 4.2 illustrates that for a given $\mathcal{D}$, $\mathcal{H}_t$ is trained using the truncated training set $\mathcal{D}_t$, at each time point t. Thus, at every time point, V number of classifiers are learned. If the number of time points are T, then total $V * T$ PCs are learned from the $\mathcal{D}$. Further, $\mathcal{H}_t$ is used to get the posterior probabilities of unlabelled MTS at any given time point t.

***Step 2**: ESRs definition*:

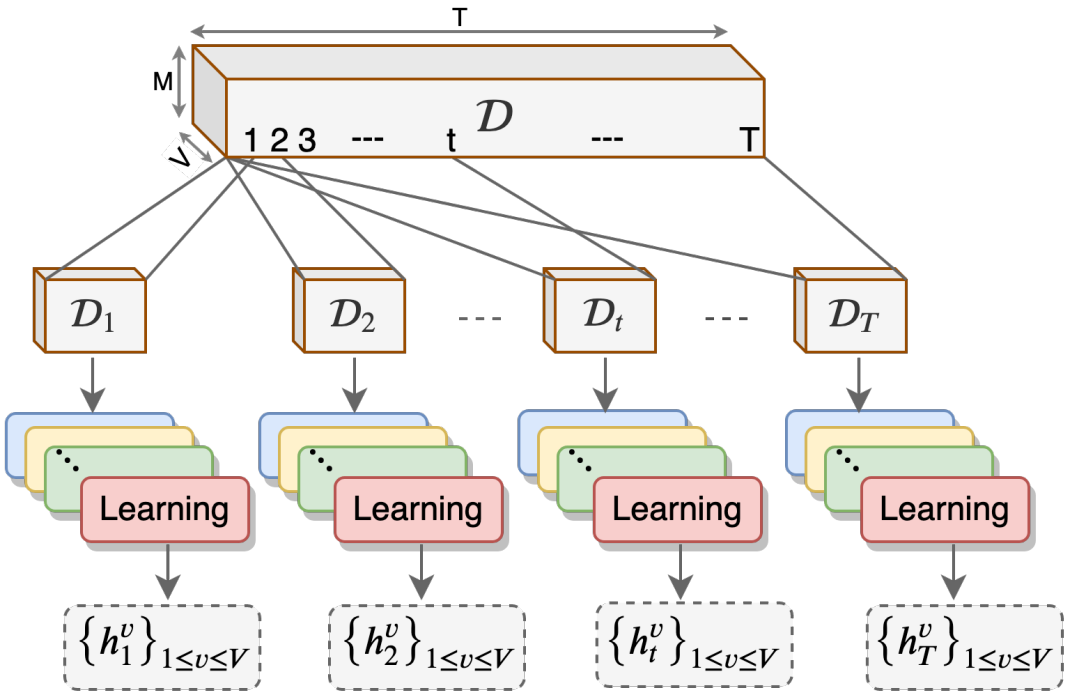


Figure 4.2: Training a set of classifiers for MTS

ESRs are one of the vital steps in the proposed early classification model for MTS, since they provide support in the decision making process of early classification. In the proposed model, two ESRs ($\mathcal{R}1_\Theta$ and $\mathcal{R}2_\Theta$) are defined, and the first ESR $\mathcal{R}1_\Theta$ is expressed as

$$\mathcal{R}1_\Theta(\Pi^t, t) = \begin{cases} 0 & \text{if} \quad \Theta_0(\frac{t}{T}) + \sum_{v=1}^{V} \Theta_{2(v-1)+1}\Pi^t_{v,1} + \Theta_{2(v)}(\frac{\Pi^t_{v,1}}{\Pi^t_{v,2}}) \le 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.1}$$

where $\Theta = (\Theta_0, \Theta_1, \ldots, \Theta_{2V})$ is a vector of parameters of $\mathcal{R}1_\Theta$. Each parameter takes the real value between -1 and 1. The parameters are learned through the optimization process, which is discussed in *step 4*. $\Pi^t \in \mathbb{R}^{V\times K}$ is the set of posterior

probabilities of a MTS X at time t, returned by corresponding $\mathcal{H}_t$, where K is the number of classes in the dataset. $\Pi^t_{v,1}$ and $\Pi^t_{v,2}$ denote the first and second highest posterior probabilities of v^{th} variable of MTS at time t.

The ESR, defined in Eq.4.1, consists of three components: (i) The ratio of current time point t and length of the series T, (ii) The highest probability of each variable of MTS, and (iii) The ratio of highest and second-highest probabilities of each variable of MTS. The first component is included to support the earliness factor in the decision process. Because, as time t progresses, the corresponding delaying cost increases. The last two components are utilized to assist the reliability of the decision. If the ratio of the two highest class probabilities is more, then the prediction will be more reliable.

The second ESR $\mathcal{R}2_{\boldsymbol{\Theta}}$ extends the intuition of SR2 [11] for MTS. The ESR $\mathcal{R}2_{\boldsymbol{\Theta}}$ considers all the class probabilities for each variable in MTS X, return by $\mathcal{H}_t$ at time t. ESR $\mathcal{R}2_{\boldsymbol{\Theta}}$ is formally defined as

$$\mathcal{R}2_{\Theta}(\Pi^t, t) = \begin{cases} 0 \quad \text{if} \quad \Theta_0\left(\frac{t}{T}\right) + \sum_{v=1}^{V} \Theta_{K(v-1)+1}\Pi^t_{v,1} + \Theta_{K(v-1)+2}\Pi^t_{v,2} + \\ \qquad\qquad\qquad\qquad\qquad\qquad \cdots + \Theta_{K(v-1)+K}\Pi^t_{v,K} \leq 0 \\ 1 \quad \text{otherwise} \end{cases} \tag{4.2}$$

where $\Theta = (\Theta_0, \Theta_1, \ldots, \Theta_{VK})$ is a vector of parameters and $\Pi^t_{v,k}$ is the predicted probability of v^{th} variable for k^{th} class at time point t. This $\mathcal{R}2_{\boldsymbol{\theta}}$ leverages the complete probabilistic output of an MTS for performing the decision task.

***Step 3**: Cost function definitions*:

The aim of defining the cost function is to learn the Θ parameters of ESRs. The learning of Θ depends on the shape of ESR and cost function. Therefore, the cost function includes misclassification costs and delaying decision costs to attain the objectives of

accuracy and earliness simultaneously. Moreover, the α parameter is used to assign the relative weight between these objectives. As the value of α lies between 0 and 1, there are two extreme cases. If $\alpha = 0$ then the accuracy factor becomes 0 and if $\alpha = 1$ then the earliness factor becomes 0. The proposed cost function for learning ESR is defined as

$$C(\mathcal{D}, \mathcal{R}_\Theta) = \frac{1}{m} \sum_{i=1}^{m} (\alpha\, C_{miss}(\mathbf{X}^i, \mathcal{R}) + (1-\alpha)\, C_{delay}(\mathbf{X}^i, \mathcal{R})) + \lambda\, lr(\Theta) \tag{4.3}$$

where α is a balancing parameter between accuracy and earliness, C_{miss} is misclassification cost of a MTS and C_{delay} is cost of delaying the decision to classify the MTS. λ (≥ 0) is a regularization parameter and $lr(\Theta)$ is a regularization term [94]. When $lr(\Theta) = 0$, it indicates no regularization and therefore, it is the standard cost function. In addition, both regularization l_1 and l_2 are operated in the cost function to reduce the effect of overfitting. l_1 regularization is defined by $lr(\Theta) = \|\Theta\|_1 = \sum_{j=1}^{len(\Theta)} |\Theta|$ and l_2 regularization is defined by $lr(\Theta) = \|\Theta\|_2 = \sum_{j=1}^{len(\Theta)} \Theta^2$. Thus, the variants of a cost function are denoted as C_{no}(no regularization), C_{l_1} (l1-regularization), C_{l_2} (l2-regularization) and effect of variants are analyzed in Section 4.4.4.

Delaying decision cost: The delaying decision cost increases as the number of sample data points increases. It is scaled between 0 and 1. If i^{th} MTS in training set is classified at time point t (denoted by t^*) then delaying decision cost is defined as:

$$C_{delay}(\mathbf{X}^i, \mathcal{R}) = \frac{t^*}{T} \tag{4.4}$$

Misclassification cost: It is evaluated based on (0-1) loss. If the predicted output is equal to true output, then cost is considered 0, otherwise 1. The C_{miss} for i^{th} MTS in training set is calculated as:

$$C_{miss}(\mathbf{X}^i, \mathcal{R}) = \Psi\left(\underset{v\in[1,V]}{^{majority}}\left(\underset{k\in[1,K]}{^{argmax}}\left(\Pi_{v,k}^{t^*}\right)_{v\in[1,V]}\right) \neq \hat{y}\right) \tag{4.5}$$

where,

- $\underset{k\in[1,K]}{^{argmax}}\left(\Pi_{v,k}^{t^*}\right)$ returns the class corresponding to the maximum probability of v^{th} dimension of MTS.
- $\underset{v\in[1,V]}{^{majority}}(.)$ returns the class having highest majority in voting. If the *majority voting* ties, then class is being returned, which has the highest probability among them. It is explained with example, later in this step.
- $\Psi(.)$ returns 0, if the predicted class label is equal to true class label otherwise returns 1.

Majority voting tie resolution: Lets consider two scenarios of probabilities $\Pi^{'}$ and $\Pi^{''}$ where $V = 4$ and $K = 3$.

$$\Pi^{'} = \begin{bmatrix} \mathbf{0.54} & 0.34 & 0.12 \\ \mathbf{0.84} & 0.12 & 0.04 \\ 0.35 & \mathbf{0.46} & 0.19 \\ 0.05 & 0.17 & \mathbf{0.78} \end{bmatrix}, \quad \Pi^{''} = \begin{bmatrix} \mathbf{0.54} & 0.34 & 0.12 \\ \mathbf{0.84} & 0.12 & 0.04 \\ 0.06 & \mathbf{0.89} & 0.05 \\ 0.02 & \mathbf{0.87} & 0.11 \end{bmatrix}$$

In the first scenario $\Pi^{'}$, the function *argmax* returns a class vector (1, 1, 2, 3) in which class 1 has the highest frequency of occurrence. Therefore *majority* function will return 1 as an output class. In the second scenario $\Pi^{''}$, the function *argmax* returns a class vector of (1, 1, 2, 2) in which both the classes have an equal frequency of occurrence and thus the decision will be a tie. So, the probability vector (0.54, 0.84, 0.89, 0.87) is utilized to break this tie, which contains the variable wise highest class probability corresponding to the class vector (1, 1, 2, 2). As a results, the *majority* function will return 2 as the output class label, based on the corresponding highest class probability (0.89) in the probability vector.

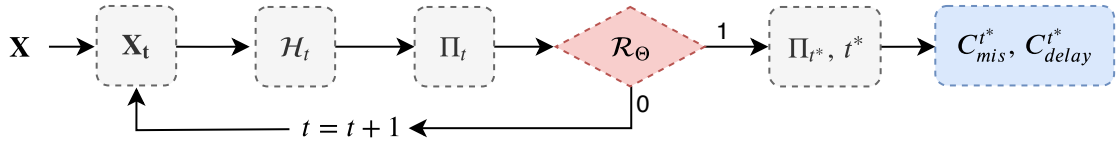


Figure 4.3: Cost evaluation process for a MTS **X**

***Step 4**: ESRs learning*:

This step explains the learning procedure of ESRs. The ESR requires two-parameters (Θ, Π) to decide whether the prediction is reliable or not at any time point t, as it have been defined in Eq. 4.1 and Eq. 4.2. This step aims to learn Θ, by minimizing the cost function, defined by Eq. 4.3 through optimization methods. The proposed cost function is non-convex and and non-differentiable [11]. Therefore, the population-based optimization methods are the best choice. Hence, PSO is selected for performing optimization exercise [97]. It is worth noting that the PSO is computationally effective, as compared to the other population-based methods such as GA [112].

In this process, C_{miss} and C_{delay} are calculated for each MTS in the training set and then mean values are recorded to compute overall cost, as defined in Eq. 4.3. This cost needs to be minimized to learn Θ. The process of cost evaluation is illustrated in Figure 4.3. Initially, t starts from 1 and passes the subsequence of **X** into corresponding ad-hoc $\mathcal{H}_t$. Further, the probabilistic output (Π_t) is passed into ESR to make a decision. If ESR returns 0 (unsatisfied) then it increments the t and repeat the process. If ESR returns 1 (satisfied), then the current time point t^* and corresponding Π_{t^*} are used to calculate C_{delay} and C_{miss} using Eq. 4.4 & Eq. 4.5 respectively.

The learning of the ESRs utilized K-fold cross-validation to overcome the over-fitting problem. This process first partitioned the training data into K-folds and then K-1 folds are used for training the ad-hoc $\mathcal{H}_t$, and remaining fold is utilized to generate the class probabilities in each iteration.

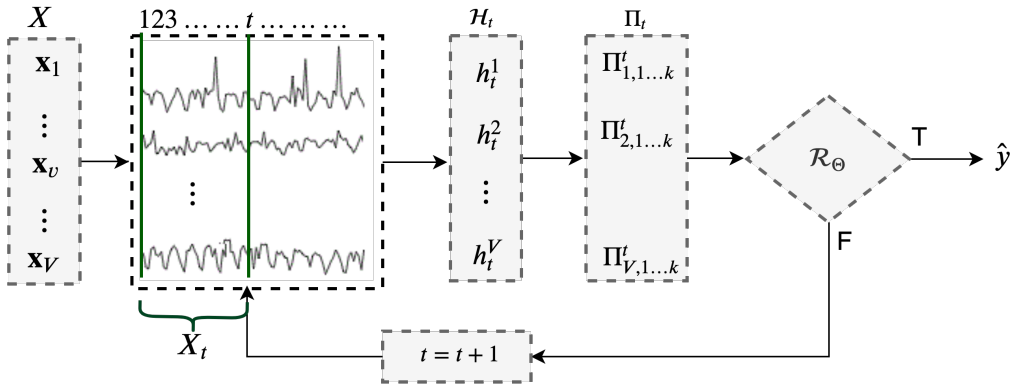


Figure 4.4: Prediction process for a incoming MTS **X**

4.3.2 Prediction phase

The proposed model has learned two components, a set of classifiers $\mathcal{H}$ and ESRs. A set of classifiers $\mathcal{H}_t$ at each time point t is trained using a complete training dataset, and ESR's are learned through the optimization process. Finally, the trained model is used for early prediction on unseen MTS, as shown in Figure 4.4. The $\mathbf{X}_t$ is provided to corresponding classifiers $\mathcal{H}_t$, which returns the class probabilities. Further these probabilities are presented to ESR. If ESR returns true, model halts and predicts the class level, or otherwise waits for more data.

4.4 Experimental evaluation

This section provides the details of evaluation methods, datasets description and various parameter settings to analyze the results. The simulation of this model is performed in R on a personal computer having Intel Core i7 processor with 3.6 GHz clock frequency and 16 GB main memory.

4.4.1 Evaluation metrics

The literature informs that there are two performance measures, generally used for early classification models. They are accuracy and earliness. As per given definition

.1

Table 4.1: Datasets description

Dataset	Min length	Max length	Classes	Variables	Train samples	Test samples
Wafer	104	198	2	6	298	896
ECG	39	152	2	2	100	100
Character Trajectories (ChT)	109	205	20	3	300	2558
CMUsubject16 (CMU16)	127	580	2	62	29	29
Libras	45	45	15	2	180	180
uWaveGesture Library (UWave)	315	315	8	3	200	4278

of these performance measures, the accuracy value should be high and earliness value should be low to select a better performing model. However, accuracy and earliness are conflicting measures and hence the selection the best performing model is difficult. Therefore we have used one more evaluation metric called harmonic mean (HM) of accuracy and earliness, defined in Eq. 4.6.

Harmonic Mean (HM): It computes the combined score of accuracy and earliness. HM will be 1 when earliness is 0% and accuracy 100%.

$$HM = \frac{2 * (Accuracy) * (1 - Earliness)}{Accurcay + (1 - Earliness)} \tag{4.6}$$

4.4.2 Dataset description

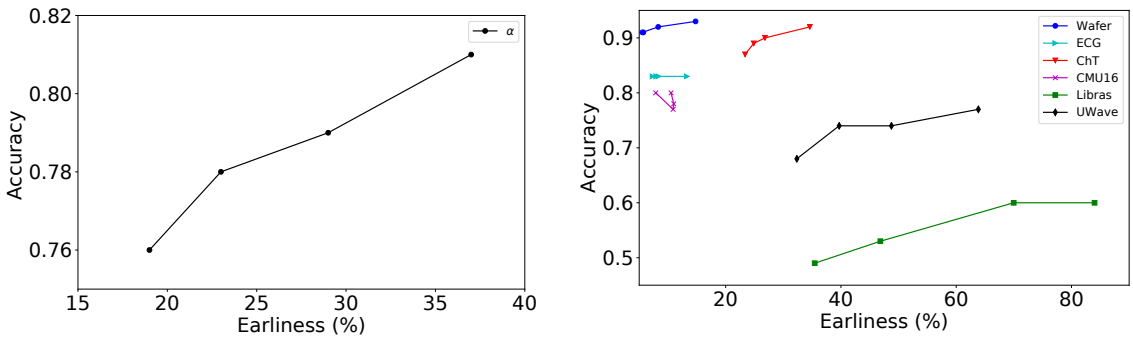
The proposed model is evaluated using six real-world publicly available MTS datasets including Wafer [113], ECG[113], Character Trajectories [45], Libras [45], CMUsubject16 [46], and uWaveGestureLibrary [24]. Moreover, we considered the pre-specified training and testing sets of these datasets from Baydogan's archive [46]. The considered datasets are diversified in nature. Therefore, the number of classes ranges from 2 to 20,

and the number of variables ranges from 2 to 62. In the pre-processing step, initially, z-score normalization is performed on each MTS in the dataset. A detailed description of the datasets is provided in Table 4.1.

4.4.3 Parameter selection

Firstly, we need to define the set of time points at which classifiers are trained. We have used different types of datasets in our experimental work. These datasets have variable length MTS, which varies from 45 to 580. Therefore, 20 equidistant time points have been considered at an interval of $(\frac{T}{20})$, where T is the length of complete MTS. Next, the cost function defined in the proposed model requires two parameters α and λ. In this experiment, the four values of α have been considered as 0.6, 0.7, 0.8, and 0.9. The value of α is considered above 0.50 to give more weight to accuracy. The effect of these α values are analyzed in Section 4.4.4.2. The value set for regularization parameter λ is {0.001, 0.003, 0.01, 0.03, 0.3, 0.1, 1,3}. Further, to learn the ESR parameters, the optimization method PSO [99] is used by considering population size 100, max iteration 100 and inertia weight 0.9. The PSO follows stochasticity and therefore we took fifteen iterations for each combination of α and λ. The result of λ is considered corresponding to the median of cost function values in all fifteen iterations, similar to [11]. For given α and Θ the parameters, ESR is considered corresponding to the best results for λ. Finally, the probabilistic classifier GP [98] has been utilized with inner product kernel and convergence threshold (1e$-$8).

In the proposed model, GP classifier considers distance-based features as input, in place of raw time series, which has demonstrated good performance in [60], [50]. The distance feature vector of a raw time series contains the pairwise distance from all the time series in the training set. For example, at a particular time step t, the train and test sets are defined as $\mathcal{D}_{v,t}^{train} \in \mathbb{R}^{m \times t}$, $\mathcal{D}_{v,t}^{test} \in \mathbb{R}^{n \times t}$ respectively, where v represents the v^{th} dimension of MTS, m and n represents the number of samples in train and test sets.



(a) (b)

Figure 4.5: Effect of α parameter : (a) Scattered plot between Accuracy and Earliness by taking average over all the datasets (b) Earliness vs. Accuracy plot for individual dataset.

Then $\mathcal{D}_{v,t}^{train}$ and $\mathcal{D}_{v,t}^{test}$ are transformed into distance-based features matrix $P_{v,t} \in \mathbb{R}^{m \times m}$ and $Q_{v,t} \in \mathbb{R}^{n \times m}$ respectively. In matrix $P_{v,t}$, $P_{v,t}[i,j]$ represents the distance measure between i^{th} and j^{th} sample in train set. In matrix $Q_{v,t}$, $Q_{v,t}[i,j]$ represents the distance measure between i^{th} sample in the test set and j^{th} sample in the train set. In this experiment, the standard Euclidean distance measure is considered to transform raw time series into the distance-based feature vector.

4.4.4 Results analysis

This section provides the analysis of the experimental results on six real-world datasets considering different parameter settings of the proposed model. Furthermore, the results are compared with traditional methods as well as other methods for early classification on MTS.

4.4.4.1 Effect of parameter α

The trade-off between accuracy and earliness is achieved through α parameter by assigning the relative weight to each component in cost function. Figure 4.5(a) plots the average value of accuracy and earliness for $\alpha \in (0.6, 0.7, 0.8, 0.9)$. It is observed that the accuracy and earliness values are increasing with α, while ranging from 0.5

to 0.9. Thus, it intuitively supports the hypothesis of cost function that higher the value of α assigns more weight to accuracy and less weight to earliness. Thus, it can be said that increasing the value of α improves the accuracy of prediction but at the same time, it also increases the average prediction time. However, it is not true for all individual datasets. It can be visualized in Figure 4.5(b) that the accuracy is not improved for *ECG* dataset while increasing the value of α from 0.6 to 0.9. It is because very few data points are sufficient to achieve good accuracy in the case of an ECG dataset. Moreover, increasing more data points does not improve the accuracy. It can be observed clearly from Figure 4.6, where the accuracy trend of the ECG dataset does not show much improvement in accuracy by increasing the number of data points in the series. However, it can be seen that the accuracy improves for other datasets e.g., *Libras*, *UWave*. For *Libras* dataset, accuracy improves from 0.49 to 0.60 (11%) and for *UWave* dataset, accuracy improves from 0.68 to 0.77 (9%) for the value of α changing from 0.6 to 0.9. Similar effect is also visible on earliness. As observed in Figure 4.5(b), the rate of change in earliness for *Libras* dataset 49% (35%-84%) is higher compared to *ChT* dataset 12% (23%-35%) with increasing the value of α from 0.6 to 0.9. Further, it has been analysed that the above changes in the behaviour of the accuracy and earliness measures depend on the accuracy pattern of individual datasets.

The accuracy pattern of datasets has been demonstrated in Figure 4.6. These patterns can be categorised into three groups. In the first category, the accuracy gradually increases with increase in the series length, as seen for *Libras*, *UWave* and *ChT* datasets. For *Libras* dataset, the accuracy changes from 0.28 to 0.64. Similarily for *UWave* and *ChT* datasets, the accuracy improves from 0.34 to 0.80 and from 0.44 to 0.94 respectively. This denotes the general convention that adding more data point in the series improves the accuracy. In the second category, the accuracy becomes stable after an interval of data points in the series. Furthermore, the addition of more data points does not significantly improve accuracy e.g., *ECG* and *Wafer* datasets. It may be possible

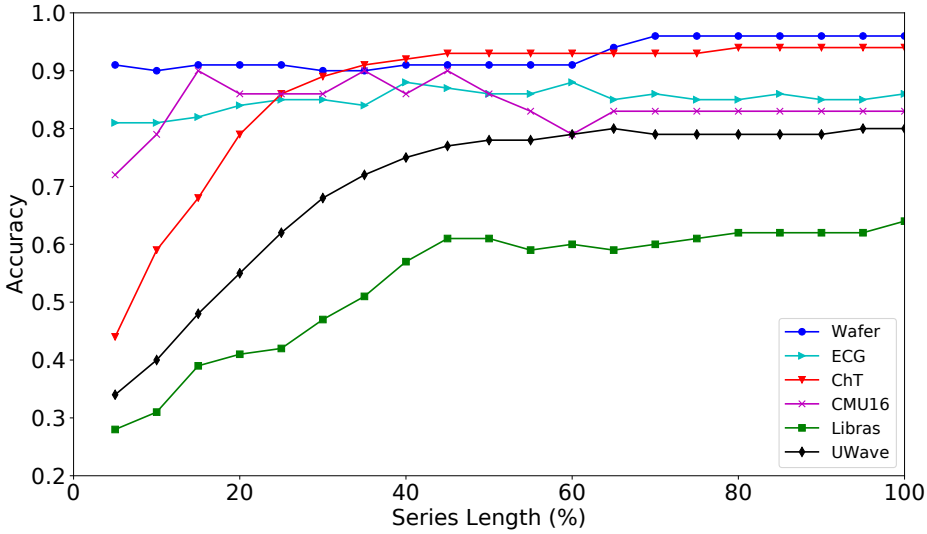


Figure 4.6: Accuracy plot at increasing length of MTS on different datasets.

that after some time points, added data points in time series are either redundant or not informative. In the third category, the accuracy trend becomes unstable or shows unusual behavior, for example *CMU16* dataset. The accuracy on this dataset increases upto 20% of the series length. After that, unstable trends are visible up to 60% of the series length before it becomes stable. Further, it has been observed that the accuracy pattern of datasets also influenced the α trends. As it can be seen in Figure 4.5(b), the first category datasets such as *UWave* and *Libras*, have shown high rate of change in accuracy as well as in earliness when α changes from 0.6 to 0.9. Whereas, the second category datasets such as *Wafer* and *ECG* have shown small changes in values of the accuracy and earliness, as compared to the first category of datasets. However, the user can choose any value of α between 0 and 1, as per need.

Furthermore, the behaviour of α is also analyzed over the combinations of ESRs and cost function which is shown in Figure 4.7. It displays the box plot for accuracy and earliness parameters over six datasets. The dots in this figure indicate the extreme low or high accuracy value, obtained with respect to one of the datasets. It has been noted that earliness is gradually increasing for all the combinations of ESRs and cost function

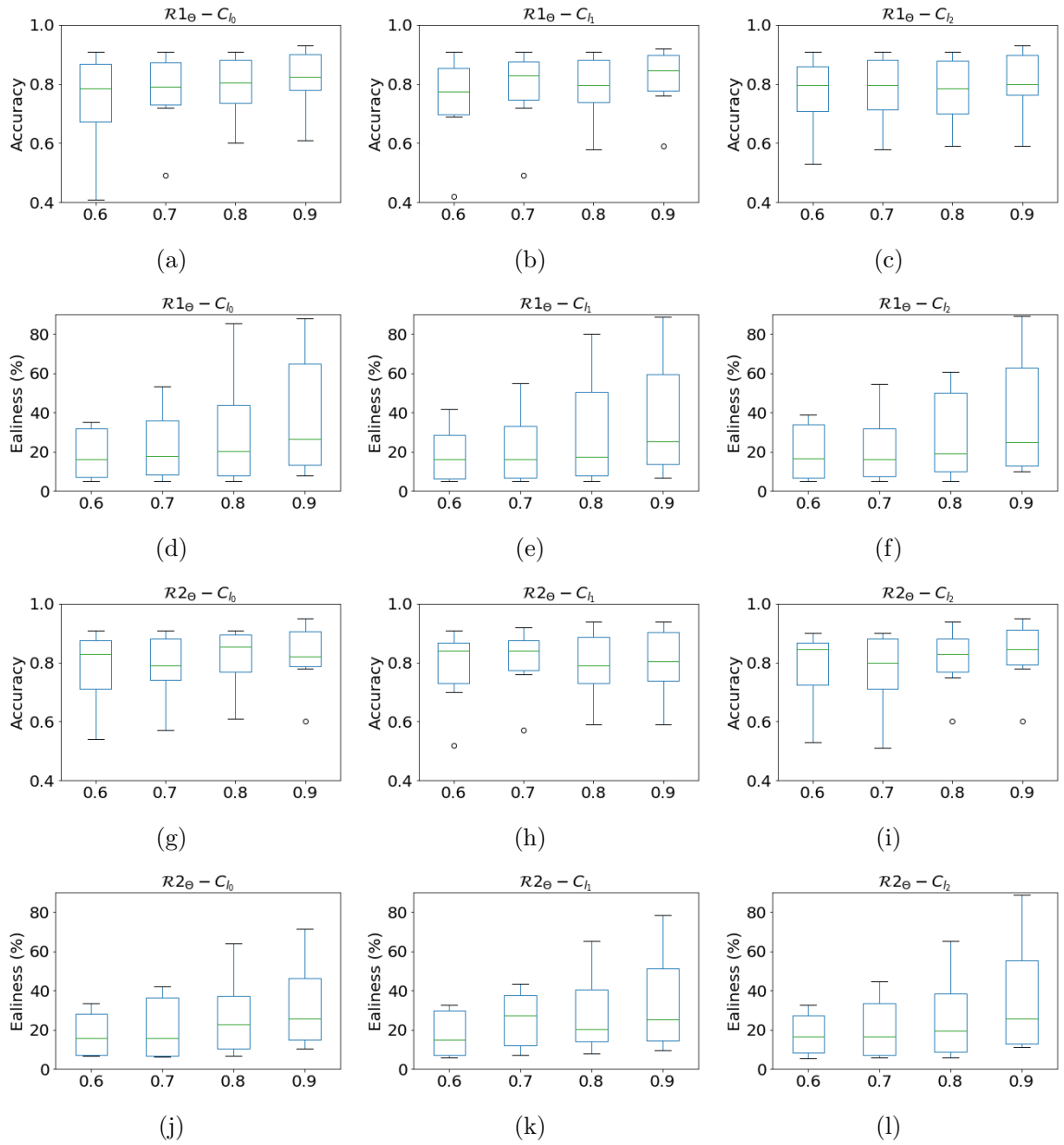


Figure 4.7: Effect of α parameter on different combination of ESRs and cost function.

while changing α from 0.6 to 0.9, except $\mathcal{R}2_\Theta - C_{l_1}$. As shown in Figure 4.7(k), median of earliness at $\alpha = 0.7$ is higher than $\alpha = 0.8$. However, this behaviour of accuracy is a little bit different for different combinations of ESRs and cost function. $\mathcal{R}1_\Theta - C_{l_1}$ displays the best accuracy on $\alpha = 0.9$ as compared to $\alpha = 0.8$ as shown in Figure 4.7(b). Moreover, $\mathcal{R}2_\Theta - C_{l_1}$ and $\mathcal{R}2_\Theta - C_{l_2}$ display similar performance for $\alpha = 0.8$

and 0.9 , which can be seen in Figure 4.7(h) and 4.7(i) respectively. Based on the above observations, it is notified that 0.8 is the more appropriate value of α to give balancing trade-off between accuracy and earliness.

4.4.4.2 Effect of regularization

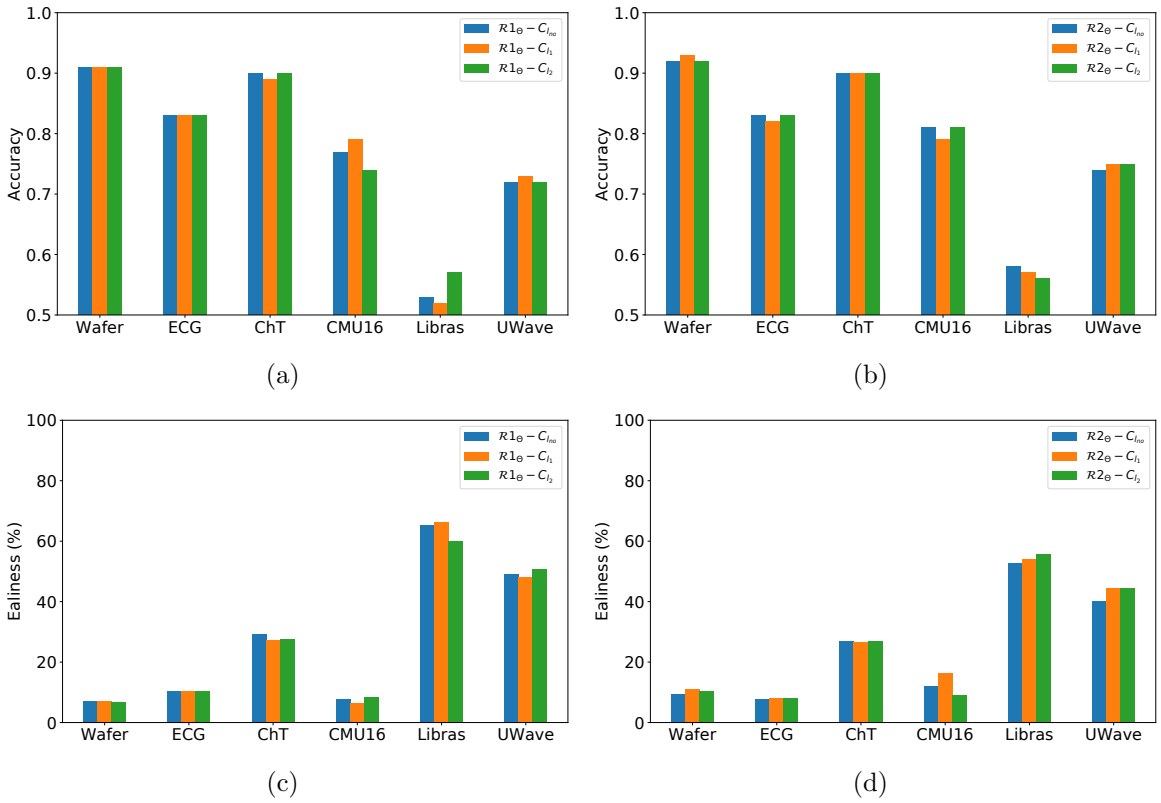


Figure 4.8: Regularization effect on ESRs $\mathcal{R}1_\Theta$ and $\mathcal{R}2_\Theta$

In this section, the effect of different variants of cost function C_{l_0}, C_{l_1} and C_{l_2} are analyzed. Figure 4.8 illustrates the average accuracy and earliness over the different values of α. It is seen that the $\mathcal{R}1_\Theta$ with C_{l_2} improves both the accuracy and earliness on *Libras* dataset, as shown in Figure 4.8(a) and 4.8(c), respectively. $\mathcal{R}1_\Theta$ with C_{l_1} slightly improved the results as compared to C_{no} and C_{l_2} on *CMU16* as well as on *UWave* datasets. Moreover, no significant effect of regularization has been observed on *ECG* and *Wafer* datasets. $\mathcal{R}2_\Theta$ with C_{l_2} improves the earliness on *CMU16* by

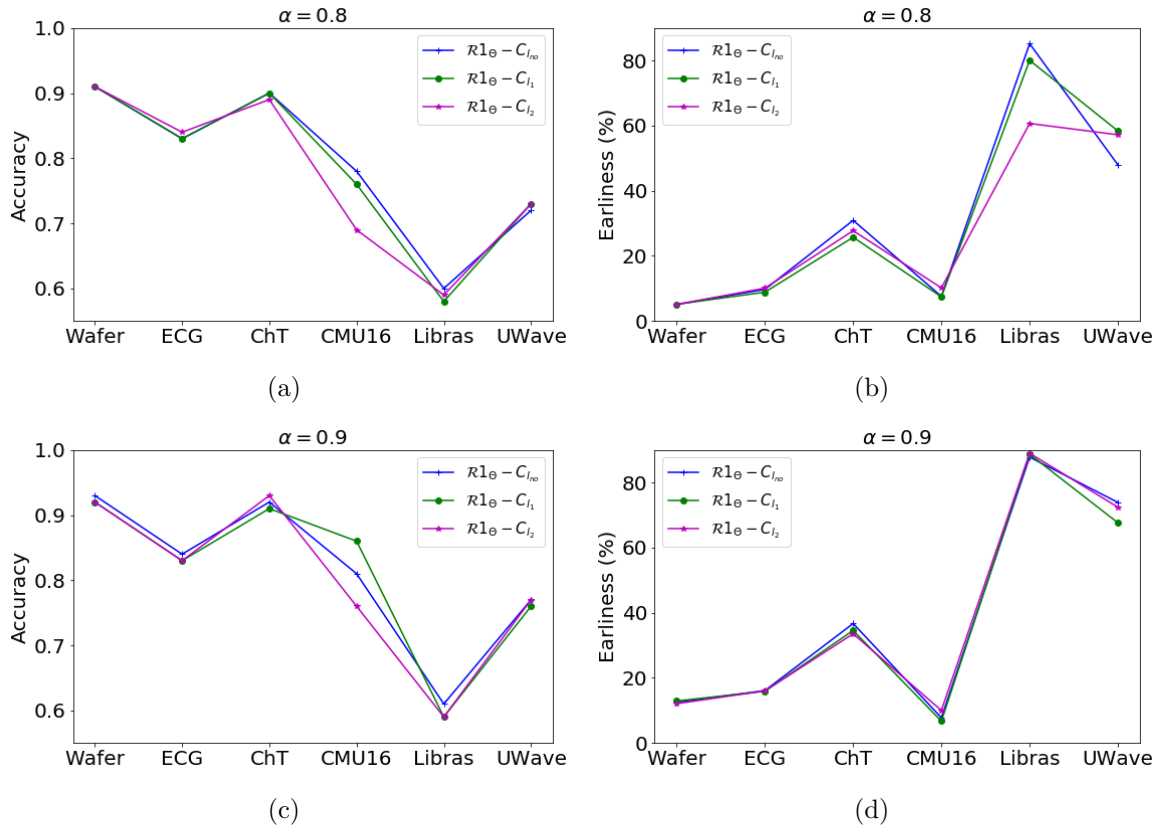


Figure 4.9: Accuracy and earliness plot of ESR $\mathcal{R}1$ for $\alpha \in \{0.8, 0.9\}$

maintaining similar accuracy with C_{no}, as shown in Figure 4.8(b) and 4.8(d). Hence, the above analysis indicates that the ESRs $\mathcal{R}1_\Theta$ and $\mathcal{R}2_\Theta$ with regularization improve the results on *ChT, CMU16, Libras* and *UWave* datasets.

In addition, the analysis of ESRs with all variants of cost function are given for $\alpha \in \{0.8, 0.9\}$. Figure 4.9 demonstrates the performance of $\mathcal{R}1_\Theta - C_{l_0}$, $\mathcal{R}1_\Theta - C_{l_1}$, $\mathcal{R}1_\Theta - C_{l_2}$, $\mathcal{R}2_\Theta - C_{l_0}$, $\mathcal{R}2_\Theta - C_{l_1}$ and $\mathcal{R}2_\Theta - C_{l_2}$ over six datasets. It is observed that none of the combinations shows its superiority over all the datasets. However, it is noticeable over individual datasets. As shown in Figure 4.9(a) and 4.9(b), $\mathcal{R}1_\Theta - C_{l_2}$ provides the best earliness on *Libras* dataset while attaining similar accuracy with $\mathcal{R}1_\Theta - C_{l_0}$ and $\mathcal{R}1_\Theta - C_{l_1}$. Whereas, $\mathcal{R}1_\Theta - C_{l_0}$ attains the best earliness on *UWave* dataset. For $\alpha = 0.9$, $\mathcal{R}1_\Theta - C_{l_1}$ achieves the highest accuracy on *CMU16* dataset and lowest earliness value on *UWave* dataset as shown in Figure 4.9(c)-4.9(d). It shows that

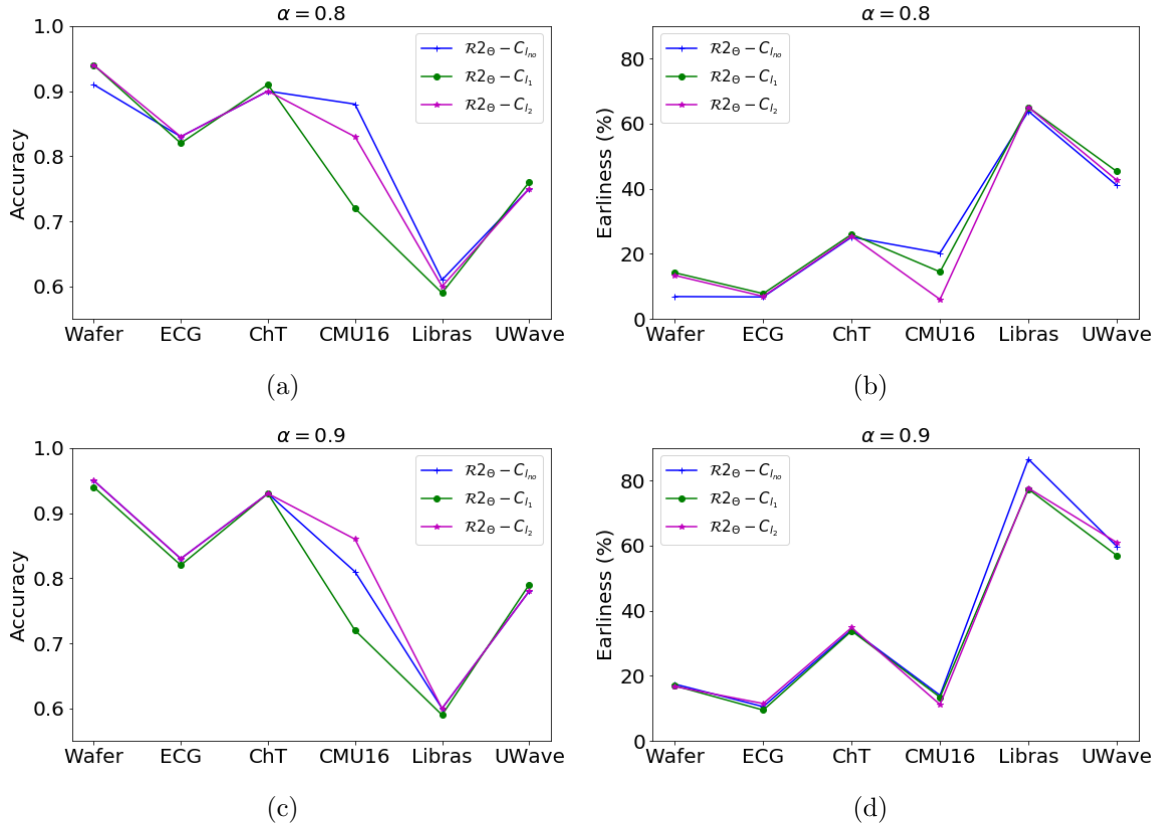


Figure 4.10: Accuracy and earliness plot of ESR $\mathcal{R}2$ for $\alpha \in \{0.8, 0.9\}$

$\mathcal{R}1_\Theta$ with C_{l_1} outperforms in one of the objectives without degrading other. $\mathcal{R}2_\Theta$ with regularization provides more balanced performance as compared to no regularization. As clearly demonstrated in Figure 4.10(a) and Figure 4.10(b), for *CMU16* dataset, $\mathcal{R}2_\Theta - C_{l_0}$ is the best in terms of accuracy but worst in terms of earliness. Similarly, on *Wafer* dataset, $\mathcal{R}2_\Theta - C_{l_0}$ is best in terms of earliness but worst in terms of accuracy. In Figure 4.10(c)-4.10(d), $\mathcal{R}2_\Theta - C_{l_1}$ and $\mathcal{R}2_\Theta - C_{l_2}$ achieve the best accuracy and earliness on *CMU16* and *UWave* datasets respectively. Thus, based on above observations, it is concluded that ESRs with regularization provide more balanced performance.

4.4.4.3 Comparison to other methods

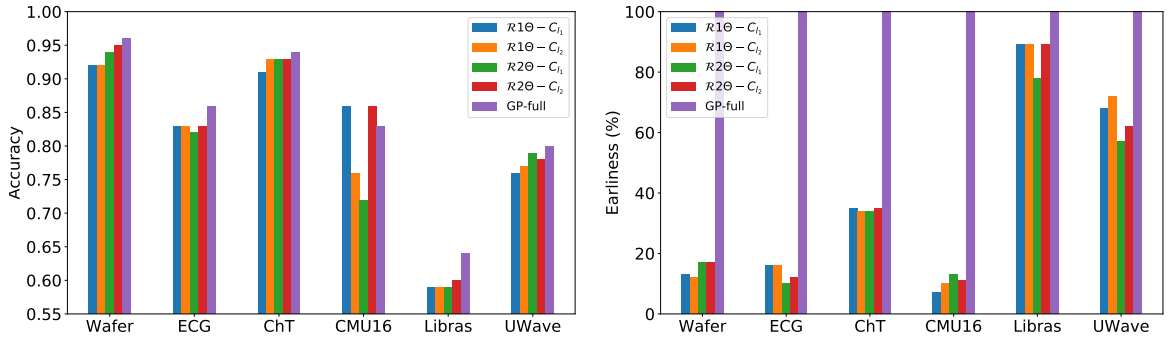
To validate the proposed model, MCFEC[59] and MTSECP [61] methods are used for comparison by considering three real MTS datasets as shown in Table 4.2. For

Table 4.2: Comparison of proposed models with other methods

Method	ECG			Wafer			ChT		
	Acc	**Ear**	**HM**	**Acc**	**Ear**	**HM**	**Acc**	**Ear**	**HM**
MCFEC-QBC[59]	77	24	0.76	90	23	0.83	—	—	—
MCFEC-Rule[59]	78	26	0.76	97	27	0.83	—	—	—
MTSECP[61]	94	54	0.62	98	57	0.59	97	70	0.46
$\mathcal{R}1_\Theta$-$\mathcal{C}_{l_1}$	82	8	0.87	94	14	0.90	91	26	**0.82**
$\mathcal{R}1_\Theta$-$\mathcal{C}_{l_2}$	83	7	**0.88**	94	13	0.90	90	25	0.81
$\mathcal{R}2_\Theta$-$\mathcal{C}_{l_1}$	83	9	0.87	91	5	**0.93**	90	26	0.81
$\mathcal{R}2_\Theta$-$\mathcal{C}_{l_2}$	84	10	0.87	91	5	**0.93**	89	28	0.80

comparative study, results for MCFEC and MTSECP are taken from original source. Moreover, the value of earliness for MCFEC method has be converted according to the given definition in the proposed model. On *ECG* dataset, the proposed method has performed better than MCFEC, in terms of both accuracy and earliness. On the other hand, MTSECP provides better accuracy than the proposed model. Moreover, on the *Wafer* dataset, the proposed model outperformed the other methods in terms of earliness and scored comparable accuracy. Further, the comparison based on *HM* metric shows that the proposed model is the best performing one compared to others. It is observed that MCFEC beats the MTSECP with 12% and 24% margin on *ECG* and *Wafer* datasets respectively. Whereas, the proposed model scores in *HM* with high marginality about 11%, 10% and 36% on *ECG*, *Wafer* and *ChT* datasets, respectively compared to other methods. It has also been noticed that the MTSECP is more centric towards accuracy and poorly centric towards earliness for all the datasets. It can be concluded that the proposed model provides a decent performance by balancing between accuracy and earliness.

Besides this comparative analysis on three datasets *ECG*, *Wafer* and *ChT*, the detailed experimental results are provided in Table 4.3. This table presents the accuracy and earliness values of all the variations of the proposed model over six datasets. $\mathcal{R}1_\Theta$-C_{l_1} on *Wafer* dataset, archives the accuracy value 0.91 for $\alpha = \{0.6, 0.7, 0.8\}$ with earliness about 5.12%. At $\alpha = 0.9$ $\mathcal{R}1_\Theta$-C_{l_1} gets the accuracy value 0.92 with earliness



(a) (b)

Figure 4.11: Comparison of the proposed model with GP-full by considering $\alpha = 0.9$

12.90%. Thus, for $\mathcal{R}1_\Theta$-C_{l_1}, $\alpha = 0.9$ is not a good choice on *Wafer* dataset. However, for $\mathcal{R}2_\Theta$-C_{l_2}, $\alpha = 0.9$ is a good choice on *Wafer* dataset as it provides the accuracy value 0.95 with earliness of 16.75%. On *ECG* dataset, all variants of the proposed model record similar performance in terms of accuracy and earliness. In contrast, the unusual behaviour of α is perceived on *CMU16* dataset. As it is seen in Table 4.3 $\mathcal{R}2_\Theta$-C_{l_1} shows best performance in terms of accuracy and earliness both at $\alpha = 0.6$, while $\mathcal{R}2_\Theta$-C_{l_2} gets best accuracy 0.86 at $\alpha = 0.9$ and best earliness 6.03 at $\alpha = 0.8$. This unusual behaviour of α on *CMU16* is directly influenced by the accuracy pattern of the dataset, as shown in Figure 4.6. As a result, it can be summarized that the variant of the proposed model provides best result at $\alpha = 0.9$ for *Wafer, ChT* and *UWave* datasets whereas for *ECG* and *Libras* datasets at $\alpha = 0.8$. On *CMU16*, the selection of α varies among the variants of the proposed model. However, selection of α completely depends on the user's need.

4.4.4.4 Comparison of the proposed model with GP-full

In this section, the different variations of the proposed model are compared with the traditional approach. Figure 4.11(a) and 4.11(b) illustrate the accuracy and earliness of various datasets. GP-Full indicates the GP classifier which is trained using full-length time series as per the conventional classification approach. Figure 4.11 shows that

Table 4.3: Accuracy and earliness of proposed model over six MTS datasets

		wafer		ECG		ChT		CMU16		Libras		UWave	
	α	**Acc**	**Ear**	**Acc**	**Ear**	**Acc**	**Ear**	**Acc**	**Ear**	**Acc**	**Ear**	**Acc**	**Ear**
$\mathcal{R}1_\Theta$-$\mathcal{C}_{l_1}$	**0.6**	0.91	5.07	0.83	8.30	0.86	24.02	0.72	5.69	0.42	41.81	0.69	29.94
	0.7	0.91	5.13	0.83	8.25	0.89	24.26	0.83	6.38	0.49	55.06	0.72	36.19
	0.8	0.91	5.14	0.83	8.85	0.90	25.72	0.76	7.41	0.58	80.06	0.73	58.39
	0.9	0.92	12.90	0.83	15.90	0.91	34.56	0.86	6.72	0.59	88.69	0.76	67.65
$\mathcal{R}1_\Theta$-$\mathcal{C}_{l_2}$	**0.6**	0.91	5.08	0.83	7.90	0.87	25.04	0.76	6.21	0.53	36.64	0.69	38.92
	0.7	0.91	5.12	0.83	8.10	0.90	24.08	0.76	7.07	0.58	54.44	0.70	34.36
	0.8	0.91	5.11	0.84	10.10	0.89	27.73	0.69	10.17	0.59	60.67	0.73	57.20
	0.9	0.92	12.08	0.83	16.00	0.93	33.55	0.76	10.00	0.59	89.00	0.77	72.43
$\mathcal{R}2_\Theta$-$\mathcal{C}_{l_1}$	**0.6**	0.92	7.14	0.83	7.10	0.87	22.16	0.86	7.41	0.52	32.69	0.70	32.03
	0.7	0.91	5.80	0.82	7.80	0.88	24.48	0.86	30.00	0.57	39.92	0.76	43.30
	0.8	0.94	14.21	0.82	7.80	0.91	26.00	0.72	14.48	0.59	65.03	0.76	45.32
	0.9	0.94	16.96	0.82	9.50	0.93	33.81	0.72	13.45	0.59	76.36	0.79	55.97
$\mathcal{R}2_\Theta$-$\mathcal{C}_{l_2}$	**0.6**	0.90	5.25	0.83	7.05	0.87	21.32	0.86	11.90	0.53	32.81	0.69	29.32
	0.7	0.90	5.73	0.83	6.95	0.90	25.77	0.69	7.76	0.51	36.19	0.77	44.53
	0.8	0.94	13.39	0.83	6.95	0.90	25.42	0.83	6.03	0.60	65.06	0.75	42.63
	0.9	0.95	16.75	0.83	11.50	0.93	34.82	0.86	11.21	0.60	77.67	0.78	60.97
$\mathcal{R}1_\Theta$-$\mathcal{C}_{no}$	**0.6**	0.91	5.11	0.83	6.60	0.88	23.80	0.74	8.45	0.41	35.13	0.65	34.51
	0.7	0.91	5.15	0.82	9.73	0.89	26.04	0.76	7.84	0.49	53.14	0.72	39.47
	0.8	0.91	5.07	0.83	9.70	0.90	30.91	0.78	7.59	0.60	85.18	0.72	47.98
	0.9	0.93	12.50	0.84	16.10	0.92	36.70	0.81	7.76	0.61	87.81	0.77	73.90
$\mathcal{R}2_\Theta$-$\mathcal{C}_{no}$	**0.6**	0.91	6.70	0.83	6.95	0.89	23.86	0.83	7.59	0.54	33.65	0.67	29.36
	0.7	0.91	6.40	0.83	6.95	0.90	24.59	0.74	6.21	0.57	42.32	0.75	40.30
	0.8	0.91	6.89	0.83	6.78	0.90	25.15	0.88	20.26	0.61	63.83	0.75	41.05
	0.9	0.95	17.47	0.83	10.55	0.93	34.16	0.81	14.05	0.60	86.79	0.78	59.79
GP	NA	0.96	100	0.86	100	0.94	100	0.83	100	0.64	100	0.80	100

$\mathcal{R}1_\Theta$-$\mathcal{C}_{l_1}$, $\mathcal{R}1_\Theta$-$\mathcal{C}_{l_2}$, $\mathcal{R}2_\Theta$-$\mathcal{C}_{l_1}$,and $\mathcal{R}2_\Theta$-$\mathcal{C}_{l_2}$, have achieved decent accuracies over all the datasets as compared to GP-full by utilizing approximately 37% of full-length MTS. $\mathcal{R}2_\Theta$-$\mathcal{C}_{l_2}$ achieves similar or even higher accuracy on *Wafer, ChT* and *CMU16* datasets while the average prediction lengths are 14.67%, 34.18% and 10.34% respectively. On *Libras* and *UWave* datasets, the proposed model is behind 3% and 2% respectively, in terms of accuracy. However, the proposed model requires approximately 86.18% and 64.76% length of MTS for *Libras* and *UWave* datasets, as compared to GP-full. Thus, from the above observation, It can be clearly seen that the proposed model required very fewer data points to classify the MTS as compared to the traditional time series classification approach. Also, the proposed model is able to provide very early decision on the four datasets (*Wafer, ECG, ChT, CMU16*) as compared to *Libras, UWave* datasets. As shown in Figure 4.6, accuracy on these four datasets (*Wafer, ECG, ChT,*

CMU16) improves by utilizing approximately upto 20% series length and after that, it becomes nearly stable. But for *Libras* and *UWave* datasets, accuracy improves with increasing the series length. Thus, it clearly shows that the proposed model for early classification on MTS is adaptive to the accuracy pattern of datasets and able to provide early classification with high reliability.

4.5 Summary

In this chapter, we proposed an optimization-based early classification model for MTS data by learning optimal ESRs. The ESRs examine the output of PCs and predict the class label when enough data points become available in the incoming MTS. The ESRs have been learned through PSO by minimizing the misclassification cost and delaying decision cost simultaneously. Besides, the proposed model has employed the GP classifier at each variable of MTS to capture the information independently and adopted the ensemble-based approach for assigning the class label to MTS. In the proposed model, the balancing between accuracy and earliness is obtained by parameter α that can be chosen based on the requirement of the user. The proposed model has been evaluated on publicly available datasets and has outperformed state-of-the-art methods by achieving balancing trade-off between accuracy and earliness.