



Early classification of multivariate data by learning optimal decision rules

Anshul Sharma¹  · Sanjay Kumar Singh¹

Received: 3 January 2020 / Revised: 23 May 2020 / Accepted: 16 July 2020 /

Published online: 1 August 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Early classification on time series has emerged as an active research area in the field of machine learning. It covers a wide range of applications in agriculture, medical and multimedia systems, including drought prediction, health monitoring, event detection, and many more. The early classification aims to predict the class label of a time series as soon as possible without waiting for the complete series. A critical issue in early classification is the learning of decision policy that determines the adequacy of the collected data required for reliable class prediction. It is more challenging for Multivariate Time Series (MTS) data, where the decision depends on multiple variables to achieve a trade-off between earliness and accuracy. Therefore, this work proposes an optimization-based early classification model for MTS data based on optimal decision rule learning. The proposed model adopts a two-layered approach. The first layer employs the Gaussian process probabilistic classifiers for each variable in MTS that provides the class probabilities at the successive time steps in the series. The second layer defines Early Stopping Rule (ESR) that performs the class prediction task. The ESR learns its parameters through the particle swarm optimization by simultaneously minimizing the misclassification cost and delaying the decision cost. This work has utilized publicly available MTS datasets to validate the proposed early classification model. The experimental results show that the proposed model achieves promising results in terms of accuracy and earliness compared to existing methods.

Keywords Time series · Early classification · Cost optimization · Multivariate data analysis · Multimedia data

✉ Anshul Sharma
anshul.rs.cse16@iitbhu.ac.in

Sanjay Kumar Singh
sks.cse@iitbhu.ac.in

¹ Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, 221005, Uttar Pradesh, India

1 Introduction

Time series (TS) is a temporal sequence of data points obtained from sequential measurements, recordings of natural processes, or human activities. With the proliferation of multimedia technology, more and more TS data have appeared, including an audio stream, video sequences, and sensory information. Thus, numerous applications have been benefited from TS-based data-driven approaches under the purview of data mining and machine learning [1, 17, 30]. Among these approaches, an early classification of TS is one of the emerging research topics and it investigates the possibility of early class prediction using partially observed data [26, 32]. Moreover, an early classification is highly essential in applications where the cost of data collection is high or time-bound class prediction is required. For example, in agricultural monitoring [31], timely prediction of droughts and shortage of multiple resources would enable the implementation of necessary measures for preventing famine and determining sustainable policy. In robotics and vision-based multimedia systems [1], early assessment of human activity aids to take the required steps in advance during a real-time interactive session with humans.

Early classification aims to predict the class label of TS at the earliest without waiting for full-length series as compared to traditional approaches that focus only on maximizing the accuracy regardless of earliness [2]. However, early classification raises two challenges: (i) To define a good decision policy that can determine whether the partially observed TS data is sufficient for reliable class prediction. (ii) To find the balancing trade-off between the two objectives: accuracy and earliness. In view of these challenges, several approaches have been proposed in the literature [9, 23, 25, 26, 37, 38] for early class prediction. For example, Xing et al. [37], put forth a remarkable work for early classification on time series (ECTS) by introducing the concept of the minimum required length for class prediction based on learning nearest-neighbor relationship in the training dataset. In [38] and [9], the authors presented shapelet-based approaches for early classification. The shapelets are the distinguishable features of TS. Therefore, in these methods, shapelet-threshold needs to be defined carefully to determine if TS has a specific shapelet for classification. However, if time series belonging to different class groups do not have a distinguishable pattern, then it is difficult to determine the threshold [39]. As a result, in such scenarios, the shapelet-based methods are not completely effective. Mori et al. [25] proposed the model-based approach for early classification on TS that follow the learning of probabilistic classifiers and reliability threshold. Recently in another work, Mori et al. [26] presented an optimization-based framework for early classification on TS by using a set of probabilistic classifiers and defining stopping rules for early decision making. Moreover, the stopping rules were optimized by employing Genetic Algorithms.

The methods mentioned above were designed for Univariate Time Series (UTS) and are not suitable for Multivariate Time Series (MTS) data directly. It has to be noted that the 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. Previous studies in this regard have provided only a few solutions for early classification on MTS data, including the notable work accomplished using shapelet-based methods [10, 14, 16, 22]. Ghalwash and Obradovic [10] 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. He et al. [14] pointed out the limitation of this prior work by arguing that the informative pattern in each variable of MTS (UTS) can lie in different

parts of the variable. But in MSD, all segments of multivariate shapelet should have the same starting and endpoint. To address this issue, He et al. [14] developed a method for early classification on MTS by extracting the core shapelets from each variable independently. Moreover, they defined two classifier query by committee and rule-based classifier to classify incoming MTS. Lin et al. [22] introduced a reliable early classification approach for MTS, which may contain numerical as well as categorical variables. They also ensured the accuracy stability of early classification as compared to full-length MTS. In [16], the authors introduced the confidence-based early classification on MTS with multiple interpretable rules. This method estimates the cumulative confidence on incoming MTS and classifies the MTS only when a certain requirement of confidence is satisfied. In short, these methods utilize the local shapelets as interpretable features that are extracted from MTS in the training dataset. Thus, irrespective of its interpretability, these methods demand intensive computation to extract informative shapelets [22]. Moreover, the existing approaches for early classification on MTS have not adequately addressed both the challenges of early classification problem. Hence, in this work, we propose an optimization-based early classification approach for MTS data in order to address the above mentioned challenges.

Motivation and significant contributions: To the best of our knowledge, this is the first work to address the problem of early classification on MTS from an optimization point of view. The motivations behind this work include: (i) Early classification on MTS has many real-world promising applications besides being a challenge due to its variability and complexity. (ii) Earliness in class prediction is always achieved at the cost of accuracy. Therefore trade-off between accuracy and earliness always exists. But existing early classification approaches for MTS do not optimize accuracy in prediction and earliness simultaneously for learning decision rules. Hence, in this work, we propose an optimization-based early classification model for MTS data by extending the framework [26] 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 underline classifiers. The significant contributions of this work are as follows:

- The proposed early classification model for MTS captures temporal information from each variable to make an early decision on MTS. The model follows a two-layered approach in which the first layer defines a set of probabilistic classifiers (PCs) and the second layer defines the decision rules.
- In the first layer, the model employs the majority voting scheme with tie resolution to provide the class label as well as probabilistic output to the second layer. Moreover, the probabilistic information of each class is extracted by executing PCs on each variable (dimension) of MTS separately with respect to time.
- In the second layer, the model defines two ESRs for an early classification on MTS and learns its parameters by considering the misclassification cost and delaying decision cost simultaneously in its cost function. Moreover, the particle swarm optimization method is used to learn the parameters of ESRs.
- The proposed model has been evaluated on six publicly available MTS datasets to validate its applicability.

The rest of the paper is organized as follows: Section 2 provides the basic definitions and related works. The proposed early classification model for MTS is described in Section 3, and it is followed by a brief description of the classifier as well as the optimization method, provided in Section 4. Section 5 and Section 6 present the experimental setup

and result analysis respectively. Finally, Section 7 provides the conclusion of the work along with future directions.

2 Background and related work

Definition 1 A *univariate time series* (UTS) is the ordered sequence of values, collected over time, defined as $\mathbf{x} = \langle x_1, x_2, \dots, x_t, \dots, x_T \rangle$ where $x_t \in \mathbb{R}$, $t \in [1, T]$ and T is the length of UTS. Additionally, a UTS or simply called TS, is considered a sequence, if its values are from a finite set.

Definition 2 A *multivariate time series* (MTS) is a column vector, defined as $\mathbf{X} = \langle \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_v, \dots, \mathbf{x}_V \rangle^T$ where each component of MTS $\mathbf{x}_v = \langle x_{v,1}, x_{v,2}, \dots, x_{v,t}, \dots, x_{v,T} \rangle$ is a raw vector (UTS) of length T . This MTS is V -dimensional and each variate x_v (v^{th} dimension of MTS) may be of different length.

Definition 3 A *incoming MTS* (incomplete MTS) contains the data points up to the time t ($t \leq T$) from initial point. It is defined as $\mathbf{X}_t = \langle \langle X_{1,1}, X_{1,2}, X_{1,t} \rangle, \langle X_{2,1}, X_{2,2}, X_{2,t} \rangle, \dots, \langle X_{v,1}, X_{v,2}, X_{v,t} \rangle \rangle^T$ where $X_{v,t} \in \mathbb{R}$ and $X_t \in \mathbb{R}^{V \times t}$.

Definition 4 Early classification on time series is a supervised learning task that classifies the incoming TS X when enough data points are collected in the series for reliable class prediction. The early classification model is presented in Fig. 1. The model processes the incoming X at defined time points and at each time point t , X_t is presented to classifier. Further, the output of the classifier is analyzed by the decision criterion. If the criterion is satisfied, the model predicts the class label. Otherwise the model waits for more data points to be added in the series. This process is continued until the model satisfies the decision criterion or the end of the series is encountered.

2.1 Related work

In the last few decades, various traditional time series classification methods have been studied [2], whereas, in recent times, early classification on TS data has received great research interest [12, 23, 25, 39]. Thus, several methods have been reported in the literature to address an early classification problem such as instance-based learning [5, 24, 37], shapelet-based methods [10, 14, 38], model-based approaches [7, 21, 26], and other methods [12, 18]. The majority of these prior works have considered UTS data only, and very

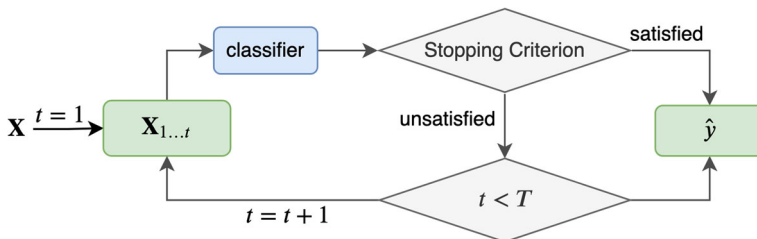


Fig. 1 Early classification model

few of them have taken MTS data into account. Initially, the methods for early classification on TS were developed by considering a fixed set of time points to train the set of classifiers and predict the class label of the test sample based on the incomplete TS [11]. Bregón et al. [5] used a case-based reasoning method for early fault classification in the laboratory plant. The method utilizes the K- nearest neighbour classifiers with different distance measures for analysis. The above mentioned methods classify the TS at prefixes of TS and as a result, no adaptive decision policy has been designed for early classification.

Xing et al. [37], formally defined the early classification of TS problem and presented a 1-NN based early classification approach that analyses the nearest neighbour stability relationship in the training set. This paper presents the concept of minimum prediction length (MPL) that learned for each TS in the training set. This approach classifies the new TS based on the MPL of matching TS in the training set. It also tries to achieve a decent early classification accuracy in comparison to the conventional 1-NN approach on full-length TS. A similar approach for MTS, namely Multivariate Time Series Early Classification based on Piecewise aggregate approximation (MTSECP), has been presented in [24] by including two additional pre-processing steps. First, MTS is converted into UTS by computing center sequence and then, dimensionality reduction is performed with the help of the piecewise aggregate approximation technique. MTSECP does not utilize variable's information in MTS effectively. These instance-based methods basically define the MRL for each TS in the training set and do not consider the earliness in their learning.

Shapelet-based methods for early classification are highly adopted, notably in the domain of medical and health informatics, due to its interpretability. Basically, shapelets are the sub-sequences of TS that have the discriminating power to differentiate among multiple classes. Moreover, shapelets represent the distinct class patterns, and hence, they are called as interpretable features. The baseline of this type of approach for early classification on UTS is presented in [38] and is denominated as Early Distinctive Shapelet Classification (EDSC). Firstly, EDSC adopted two methods, namely kernel density estimation and chebyshev's inequality, for learning the shapelet threshold. Then they mined the best local shapelets using defined utility measure, which are highly effective for early classification. Ghalwash et al. [9] proposed an extension of the EDSC with an additional uncertainty estimation property since EDSC does not have any assessment of the uncertainty while making a decision on the class prediction. Furthermore, Ghalwash and Obradovic [10] extended the concept of shapelets for early classification on MTS and proposed a method named Multivariate Shapelet Detection (MSD). The MSD extracts the local key shapelets from N-dimensional MTS in the training set and classifies the new incoming MTS based on best matching key shapelet. The limitation of this method lies in the employment of a sliding window in mining the shapelets. As an effect, all the sub-sequences in a multivariate shapelet have the same start and endpoint. However, in many realistic scenarios, the informative patterns in the variable can lie in a different part of MTS and need not be synchronous. He et al. [14] have tackled this issue and introduced a method called Mining Core Feature for Early Classification (MCFEC). The MCFEC method at first extracts the shapelets for each variable independently and then selects the core shapelets by proposing a utility measure termed generalized extended F-measure. Finally, two classification strategies were proposed (query by a committee and rule-based) to classify the incoming MTS.

Lin et al. [22] developed a reliable early classification method for heterogeneous MTS data, including categorical and numerical attributes. This method builds an early serial classifier that ensures accuracy stability compared to the full-length time series classifier. Data imbalance is a common problem in many real-world applications. In this regard, an ensemble-based early classification framework is presented in [15], called early prediction

on imbalanced MTS, that can effectively handle inter and intra class imbalance for early classification. Later, He et al. [16] extended this work by considering confidence estimation for reliable early class prediction on MTS. Recently, Zhao et al. [40] developed an early classification approach for patient monitoring in ICU. They extracted multivariate early shapelets called *MEShapelet* and predicted asynchronous MTS with interpretability. The above given literature reveals the fact that the shapelet-based methods are highly interpretable for class prediction. However, two critical issues exist with regard to this approach. Firstly, the shapelet's threshold is very hard to define if the class-wise patterns are not well distinguishable. Secondly, the process of extracting informative shapelets is highly time-consuming and complex. As a result, shapelet-based methods are computationally expensive.

A simple and effective model-based early classification approach has been presented in [25], based on discriminating the classes over time. This model develops the set of probabilistic classifiers at different timestamps and computes the reliability threshold for each class label. Moreover, the model also defines the discriminative safeguard point for each class. It classifies the incoming TS only if the reliability threshold (the difference between the two highest class probabilities) and specified safeguard point for the predicted class level are satisfied. A similar approach is adopted in [33], in which the reliability threshold is defined based on uncertainty information in class prediction. Lv et al. [23] developed a relatively similar framework, in which the confidence threshold was defined by fusing the classifier's true prediction probabilities at successive time steps. This framework is adaptable for both probabilistic as well as discriminative classifiers. A distance transformation based framework for early classification on TS was put forth by Yao et al. [39]. They transformed the TS into distance space using interpretable sub-sequences and trained the probabilistic classifiers. Finally, the confidence area was proposed as a criterion for early decision making on incoming TS. Li et al. [21] applied an early classification approach for human activity recognition, based on partially observed activity information. They modelled the 3D action recognition problem as a stochastic process called a dynamic marked point process. The early activity recognition problem was also tackled by employing the probabilistic graphical model [1]. The model captures the high-label description of the human body as skeletons from low-cost depth sensors and is able to classify the activity by considering partially observed multivariate data. Hsu et al. [18] proposed deep learning-based early classification on MTS through attention mechanism, which helped in identifying best performing segments in variables of MTS. Finally, the authors in [12] presented a meta-algorithm for early classification on MTS sensory data in the application of road surface detection. They learned the MPL class-wise and employed an ensemble-based approach that classified incoming MTS which had an unequal sampling rate of sensor's data.

The above-discussed approaches have demonstrated good results for early classification. However, they did not optimize the two conflicting objectives, i.e., accuracy and earliness, simultaneously. In this regard, Dachraoui et al. [7] proposed a meta-algorithm for early classification on TS that is non-myopic in nature. The authors included the misclassification cost and delaying decision cost in its optimization function for balancing the accuracy and the earliness. Thus, during testing, incoming TS was classified only if the estimated cost at the current timestamp is less than the estimated cost at all future timestamps. However, this method used a clustering approach for evaluating future costs, which caused a lack of clarity in the overall process. Later, Tavenard and Malinowski [35] introduced two new strategies (*NoCluster* and *2Step*) by eliminating the clustering step for further improvement. Mori et al. [26], introduced a framework for early classification on TS by defining the stopping rules as decision criteria and learned the rules by optimizing the accuracy as well as earliness

simultaneously. In this line, Recently, Sharma and Singh [34] presented an optimization-based approach for early malware detection by learning early decision rule through particle swarm optimization.

There are certain limitations in the existing approaches that have been identified from the literature discussed so far. Firstly, most of the early classification methods have been developed for UTS, with a few for MTS. Secondly, the majority of early classification approaches for MTS are the features-based methods. They are highly expensive computationally even for a moderate training set. Moreover, these methods perform well only if time series belongs to a class group, have distinct patterns from other class groups. Finally, existing methods for MTS do not consider the trade-off between the objectives accuracy and earliness, which is a desirable property for early classification problems [26, 37]. Early classification on MTS data is also more challenging due to its complexity and size, such as unequal length variables and heterogeneity of data, etc. Therefore, by considering all the above limitations, this work proposes an optimization-based early classification model for MTS data, which takes accuracy and earliness into consideration while learning decision rule. Moreover, the effect of the trade-off parameter is analyzed with several publicly available MTS datasets. A detailed description of the proposed model and results are provided in the following sections.

3 Model description

This section provides a complete description of the proposed early classification model for MTS. Figure 2 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, K-fold cross-validation process is adopted. In each iteration,

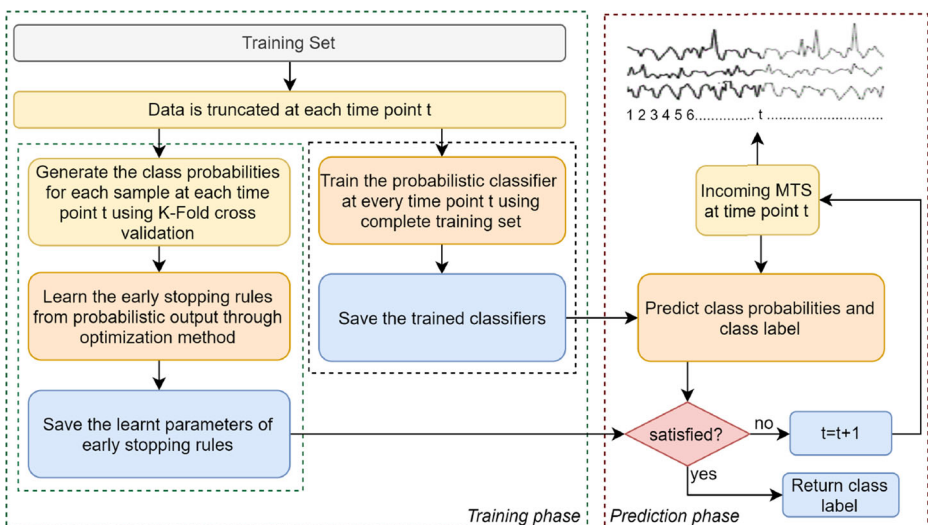


Fig. 2 Block diagram of the proposed model for early classification on MTS

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. The training steps in detail are provided in the following subsections.

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 \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 3 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:* 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

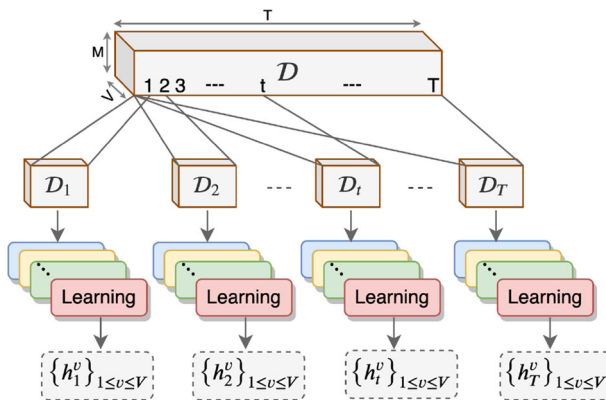


Fig. 3 Training a set of classifiers for MTS

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

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

where $\Theta = (\Theta_0, \Theta_1, \dots, \Theta_{2V})$ 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. $\Pi_{v,1}^t$ and $\Pi_{v,2}^t$ denote the first and second highest posterior probabilities of v^{th} variable of MTS at time t .

The ESR, defined in (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 $\mathcal{R}2_{\theta}$ extends the intuition of SR2 [26] for MTS. The ESR $\mathcal{R}2_{\theta}$ considers all the class probabilities for each variable in MTS X , return by \mathcal{H}_t at time t . ESR $\mathcal{R}2_{\theta}$ is formally defined as

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

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

Step 3: *Cost function (CF) 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 CF. Therefore, CF 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

$$C(\mathcal{D}, \mathcal{R}_{\Theta}) = \frac{1}{m} \sum_{i=1}^m (\alpha C_{miss}(\mathbf{X}^i, \mathcal{R}) + (1 - \alpha) C_{delay}(\mathbf{X}^i, \mathcal{R})) + \lambda lr(\Theta) \tag{3}$$

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. $\lambda (\geq 0)$ is a regularization parameter and $lr(\Theta)$ is a regularization term [27]. When $lr(\Theta) = 0$, it indicates no regularization and therefore, it is the standard CF. In addition, both regularization l_1 and l_2 are operated in the CF 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} (l_1 -regularization), C_{l_2} (l_2 -regularization) and effect of variants are analyzed in Section 6.

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:

$$C_{delay}(\mathbf{X}^i, \mathcal{R}) = \frac{t^*}{T} \tag{4}$$

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:

$$C_{miss}(\mathbf{X}^i, \mathcal{R}) = \Psi \left(\underset{v \in [1, V]}{\text{majority}} \left(\underset{k \in [1, K]}{\text{argmax}} \left(\Pi_{v,k}^{t^*} \right)_{v \in [1, V]} \right) \neq \hat{y} \right) \tag{5}$$

where,

- $\underset{k \in [1, K]}{\text{argmax}} \left(\Pi_{v,k}^{t^*} \right)$ returns the class corresponding to the maximum probability of v^{th} dimension of MTS.
- $\underset{v \in [1, V]}{\text{majority}} (\cdot)$ 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.
- $\Psi(\cdot)$ returns 0, if the predicted class is equal to true class label otherwise returns 1.

Majority voting tie resolution: Lets consider two scenarios of probabilities Π' and Π'' 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 Π' , *argmax* function returns a class vector (1,1, 2, 3). Now in the vector, 1 has the highest frequency of occurrence, and therefore *majority* function will return 1 as an output class. In the second scenario Π'' , *argmax* function returns class vector of (1, 1, 2, 2). Now, in this vector, all the classes have a similar frequency of occurrence and thus the decision is a tie. To handle this deadlock, the probability vector (0.54, 0.84, 0.89, 0.87) which contains the variable wise highest class probability corresponding to class vector (1, 1, 2, 2) is accessed. Now, in this case, *majority* function will return 2 as the output class label, based on the corresponding highest class probability (0.89) in the probability vector.

Step 4: ESRs learning: In this step, the learning procedure of ESR is discussed in detail. It requires two-parameters (Θ, Π) to make a decision (eg. whether the prediction is reliable or not) at any time point t , as defined in (1) and (2). This step aims to learn Θ , by minimizing the cost function, defined by (3) through optimization methods. The proposed CF is non-convex and non-differentiable [26]. Therefore, the

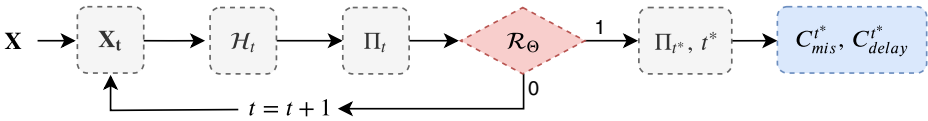


Fig. 4 Cost evaluation process for a MTS X

population-based optimization methods are the best choice. Hence, in this work, PSO is selected for performing optimization exercise [29]. It is worth noting that the PSO is computationally effective, as compared to the other population-based methods such as Genetic Algorithm (GA) [13].

In this process, C_{miss} and C_{delay} are calculated for each MTS in the training set and the average over all the MTS is recorded as defined in (3). This cost needs to be minimized to learn Θ . The process of cost evaluation is illustrated in Fig. 4. In this process, initially, t starts from 1 and passes the subsequence of X into corresponding ad-hoc \mathcal{H}_t . Further, the (Π_t) at time t is passed into ESR and 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 (4) & (5) respectively.

To bring more generality in learning the ESRs, K-fold cross-validation is used. It is also explained at the starting of this Section 3. At each time point t , training data is truncated and partitioned into K-folds. Each time K-1 folds are used for training the ad-hoc \mathcal{H}_t , and remaining fold is utilized to generate the class probabilities. Thus it aids to reduce over fitting in the learning of ESRs parameters.

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 Fig. 5. The 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.

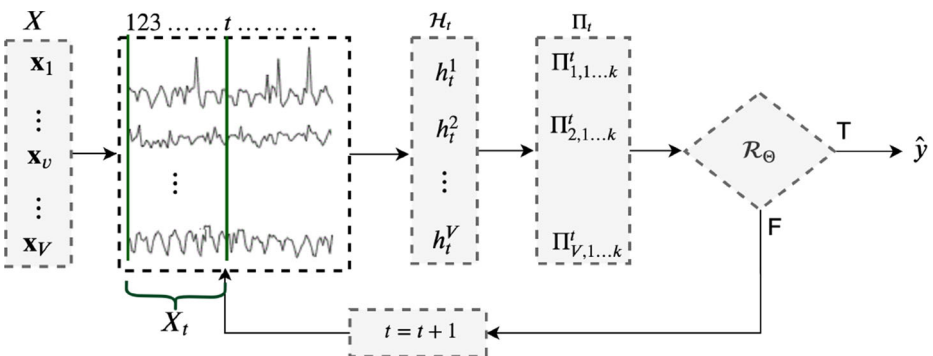


Fig. 5 Prediction process for an incoming MTS X

4 Classifier and optimization method

This section provides a brief description of Gaussian process classifier and particle swarm optimization, used underline the proposed model.

4.1 Gaussian process classifier

A Gaussian process (GP) is a collection of random variables (RVs) such that the joint distribution of every finite subset of RVs is multivariate Gaussian [36].

$$f(x) \sim \mathcal{GP}(\mathbf{m}(x), \mathbf{k}(x, x')), \quad (6)$$

where $\mathbf{m}(x)$ and $\mathbf{k}(x, x')$ are the mean and covariance functions. GPs are the natural generalization of Gaussian distributions and defined over functions. Moreover, GPs are worked as priors for bayesian inference and do not depend on training data. Basically they specify the properties of latent functions. GPs can be utilized for regression as well as for classification problem. For classification task, first GP prior is defined over latent function $f(x)$ and then the output of latent function is squashed by *logistic* or *probit* functions to obtain the prior on $\pi(x)$ which is used for class prediction [36]. For given training data $\mathcal{D} = (\mathcal{X}, \mathcal{Y}) = \{(x^i, y^i) | 1 \leq i \leq m\}$, the prediction of test sample (x^*) is naturally divided into two steps. The first is to compute the distribution of latent variable for test sample, defined as:

$$p(f^* | \mathcal{X}, \mathcal{Y}, x^*) = \int p(f^* | \mathcal{X}, x^*, \mathbf{f}) p(\mathbf{f} | \mathcal{X}, \mathcal{Y}) d\mathbf{f}, \quad (7)$$

where $p(\mathbf{f} | \mathcal{X}, \mathcal{Y}) = p(\mathcal{Y} | \mathbf{f}) p(\mathbf{f} | \mathcal{X}) / p(\mathcal{Y} | \mathcal{X})$ and then produce a probabilistic prediction by using distribution over the latent f^* , defined as:

$$\begin{aligned} \bar{\pi}^* &= p(y^* = +1 | \mathcal{X}, \mathcal{Y}, x^*) \\ &= \int \sigma(f^*) p(f^* | \mathcal{X}, \mathcal{Y}, x^*) df^*, \end{aligned} \quad (8)$$

4.2 Particle swarm optimization (PSO)

PSO is a kind of evolutionary computation technique which is used for global optimization. It is inspired by the social behaviour of fish schooling or birds flocking [29]. Moreover, PSO is population-based optimization and the group of particles in the population is known to be a swarm. Every particle in the swarm shares the information cooperatively to achieve the desired goal. Swarm particles update its path in the direction of the global best position, and the local best position, which is attained by any neighborhood particle. In this way, PSO utilizes the complete range of potential solutions, named as population set. Moreover, PSO determines the best optimal solution from the potential population via cooperation and competition.

Suppose the number of particles in the swarm is s with d -dimensional search space. Then each particle in the swarm $P = (p_1, p_2, \dots, p_s)$ is defined by d -dimensional vector for position as well as for velocity separately. The position and velocity of the i^{th} particle are represented as $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$ and $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ respectively. At every iteration, PSO updates the particle's positional vector and particle's velocity vector using (10) and (9) respectively by considering local best solution $pb_i = (pb_{i1}, pb_{i2}, \dots, pb_{id})$

and global best solution $pg_i = (pg_{i1}, pg_{i2}, \dots, pg_{id})$ [29]. The velocity and position of particles is calculated as

$$v_{i,j}^{n+1} = wv_{i,j}^n + c_1r_{1j} (pb_{i,j}^n - p_{i,j}^n) + c_2r_{2j} (pg_j^n - p_{i,j}^n) \tag{9}$$

$$p_{i,j}^{n+1} = p_{i,j}^n + v_{i,j}^{n+1} \tag{10}$$

where $i = 1, 2, \dots, s$; $j = 1, 2, \dots, d$; w is inertia weight; c_1 and c_2 are the positive real numbers, called as the acceleration constant. The variables r_1 and r_2 have the random value between 0 and 1; and $n = 1, 2, \dots$, denotes the iteration number.

5 Experimental setup

This section provides the details of evaluation methods, datasets description and various parameter settings for analyzing 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.

5.1 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 of these performance measures by (11) and (12) in the proposed work, 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 (13).

Accuracy: It is the standard measure to evaluate the performance of the model. It is defined as the ratio of truly classified MTS and the total number of MTS in the test set.

$$Accuracy = \frac{\sum_{i=1}^N (\hat{y}^i = y^i)}{N} \tag{11}$$

where N is the number of MTS in test set. \hat{y}^i is the predicted class label of i^{th} test MTS and y^i is the corresponding true class label. In (11), the value is count 1 if \hat{y}^i matched with y^i and 0 otherwise.

Earliness: It is another measure to evaluate the performance of early classification model on MTS. This is the average percentage of predicted length (t^*) to full-length of MTS, It is defined as:

$$Earliness = \frac{1}{N} \sum_{i=1}^N \frac{t_i^*}{T} \times 100 \tag{12}$$

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

$$HM = \frac{2 * (Accuracy) * (1 - Earliness)}{Accuracy + (1 - Earliness)} \tag{13}$$

5.2 Dataset description

The proposed model is evaluated using six real-world publicly available MTS datasets including Wafer [28], ECG[28], Character Trajectories [8], Libras [8], CMUsubject16 [3], and uWaveGestureLibrary [6]. Moreover, we considered the pre-specified training and testing sets of these datasets from Baydogan’s archive [3]. To show the applicability of proposed model, 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. The detailed characteristic information about the datasets is provided in Table 1.

5.3 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, twenty equidistant time points have been considered at an interval of 5%. These points are defined as 5%, 10% 15%, up to 100% of full-length TS.

Next, the CF 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 assumed above 0.50 to give more weight to accuracy. The effect of these α values are analyzed in Section 6.2. The considered 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 [4] is used by considering population size 100, max iteration 100 and inertia weight 0.9. The PSO follows stochasticity. Therefore, we took fifteen iterations for each combination of α and λ . The result of λ is considered corresponding to the median of CF values in all fifteen iterations, similar to [26]. For given α , Θ parameter of ESR is considered corresponding to the best results for λ . Finally, the probabilistic classifier GP [20] 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 [19, 25]. 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. 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 $Q_{v,t} \in \mathbb{R}^{n \times n}$ respectively. In matrix $P_{v,t}$, $P_{v,t}[i, j]$ represents the distance measure between i^{th} and j^{th} sample

Table 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

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 TS into the distance-based feature vector.

6 Results and discussion

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.

6.1 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 CF. Figure 6a 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 CF 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 Fig. 6b that the accuracy is not improved for *ECG* dataset while increasing the value of α from 0.6 to 0.9. 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 Fig. 6b, 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 Fig. 7. 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,

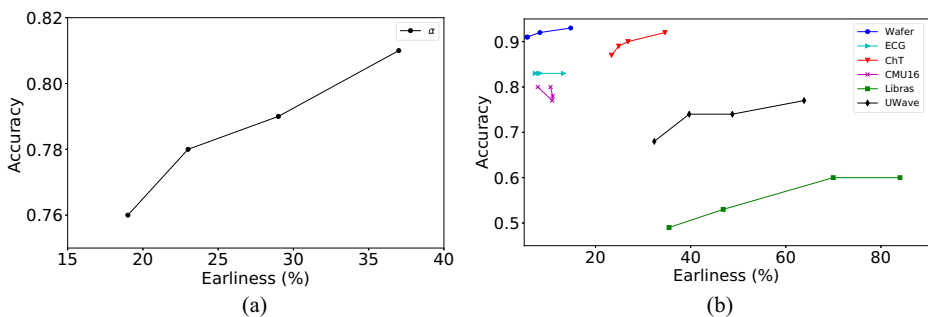


Fig. 6 Effect of α parameter : **a** Scattered plot between Accuracy and Earliness by taking average over all the datasets **b** Earliness v/s Accuracy plot for individual dataset

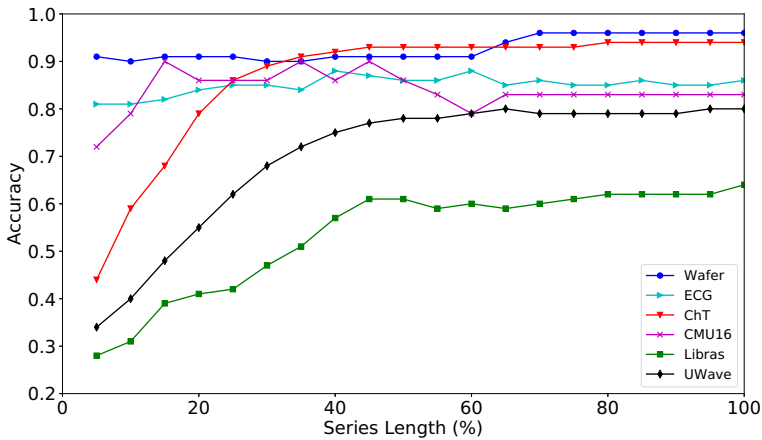


Fig. 7 Accuracy plot at increasing length of MTS on different datasets

the accuracy changes from 0.28 to 0.64. Similarly 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 that after some time points, added data points in TS 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 Fig. 6b, 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 proposed model is adaptive in nature. However, the proposed model is adaptive in nature. Therefore, based on the requirement, the user can choose any value of α between 0 and 1.

Furthermore, the behaviour of α is also analyzed over the combinations of ESRs and CF which is shown in Fig. 8. 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 CF while changing α from 0.6 to 0.9, except $\mathcal{R}2_{\ominus} - C_{l_1}$. As shown in Fig. 8k, 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 CF. $\mathcal{R}1_{\ominus} - C_{l_1}$ displays the best accuracy on $\alpha = 0.9$ as compared to $\alpha = 0.8$ as shown in Fig. 8b. Moreover, $\mathcal{R}2_{\ominus} - C_{l_1}$ and $\mathcal{R}2_{\ominus} - C_{l_2}$ display similar performance for $\alpha = 0.8$ and 0.9, which can be seen in Fig. 8h and 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.

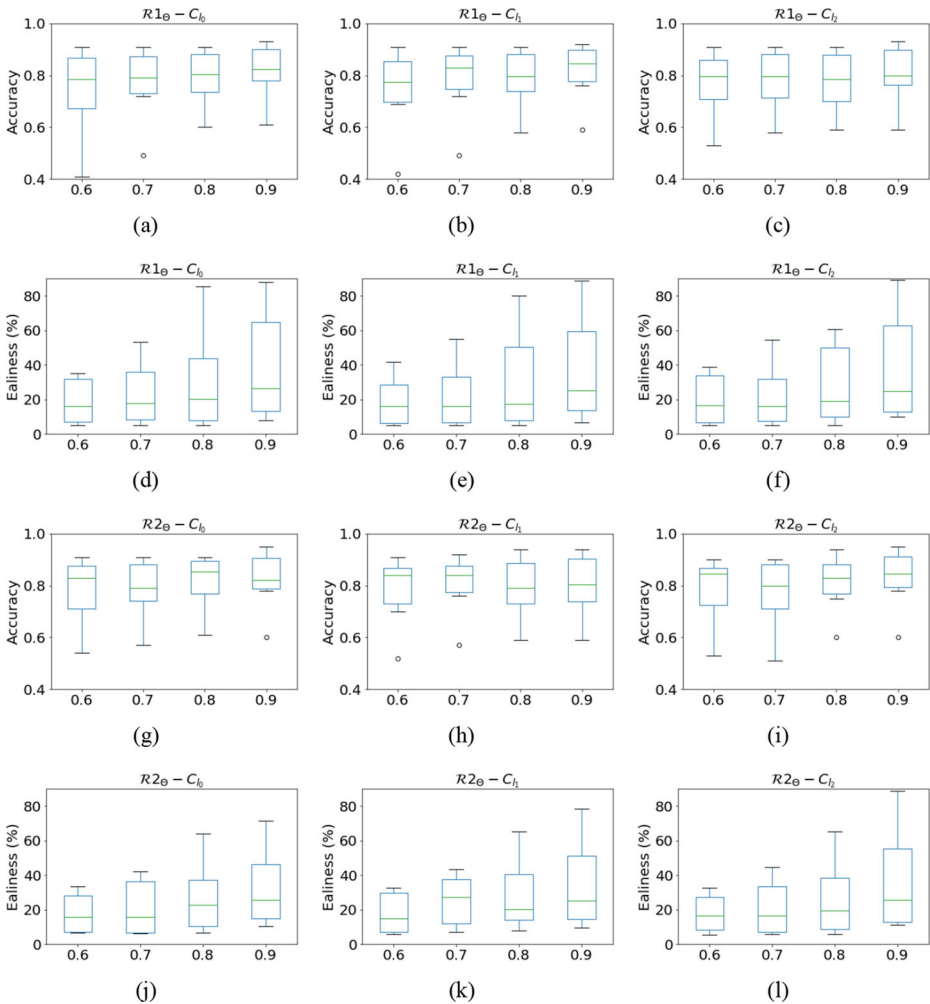


Fig. 8 Effect of α parameter on different combination of ESRs and CF

6.2 Effect of regularization

In this section, the effect of different variants of cost function C_{l_0} , C_{l_1} and C_{l_2} are analyzed. Figure 9 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 Fig. 9a and 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 maintaining similar accuracy with C_{no} , as shown in Fig. 9b and 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 CF are given for $\alpha \in \{0.8, 0.9\}$. Figure 10 demonstrates the performance of $\mathcal{R}1_{\Theta} - C_{l_0}$, $\mathcal{R}1_{\Theta} - C_{l_1}$, $\mathcal{R}1_{\Theta} - C_{l_2}$, $\mathcal{R}2_{\Theta} - C_{l_0}$,

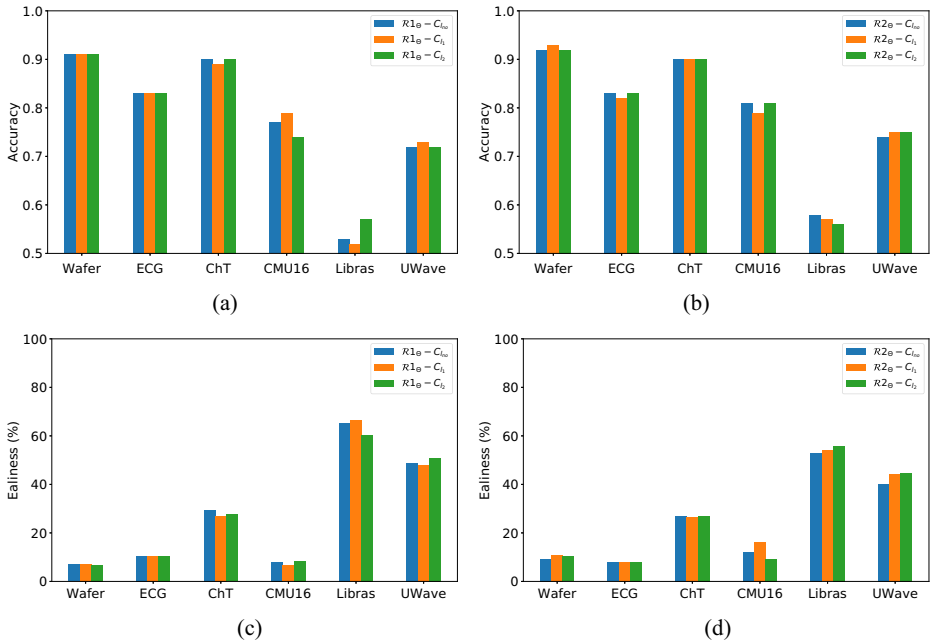


Fig. 9 Regularization effect on ESRs $\mathcal{R}1_{\theta}$ and $\mathcal{R}2_{\theta}$

$\mathcal{R}2_{\theta} - C_1$ and $\mathcal{R}2_{\theta} - C_2$ 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 Fig. 10a and b, $\mathcal{R}1_{\theta} - C_2$ provides the best earliness on *Libras* dataset while attaining similar accuracy with $\mathcal{R}1_{\theta} - C_0$ and $\mathcal{R}1_{\theta} - C_1$. Whereas, $\mathcal{R}1_{\theta} - C_0$ attains the best earliness on *UWave* dataset. For $\alpha = 0.9$, $\mathcal{R}1_{\theta} - C_1$ achieves the highest accuracy on *CMU16* dataset and lowest earliness value on *UWave* dataset as shown in Fig. 10c and d. It shows that $\mathcal{R}1_{\theta}$ with C_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 Fig. 10e and f, for *CMU16* dataset, $\mathcal{R}2_{\theta} - C_0$ is the best in terms of accuracy but worst in terms of earliness. Similarly, on *Wafer* dataset, $\mathcal{R}2_{\theta} - C_0$ is best in terms of earliness but worst in terms of accuracy. In Fig. 10g and h, $\mathcal{R}2_{\theta} - C_1$ and $\mathcal{R}2_{\theta} - C_2$ 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.

6.3 Comparison to other methods

To validate the proposed model, MCFEC[14] and MTSECP [24] methods are used for comparison by considering three real MTS datasets as shown in Table 2. For comparative study, results for MCFEC and MTSECP are taken from original source. Moreover, the value of earliness for MCFEC method has been transformed as per given definition in the proposed model. On *ECG* dataset, the proposed method has performed better than MCFEC, in terms of both accuracy and earliness. However, MTSECP provides better accuracy than the proposed model on the *ECG* dataset. Moreover, on the *Wafer* dataset, the proposed model outperforms the other methods in terms of earliness and score comparable accuracy. Further,

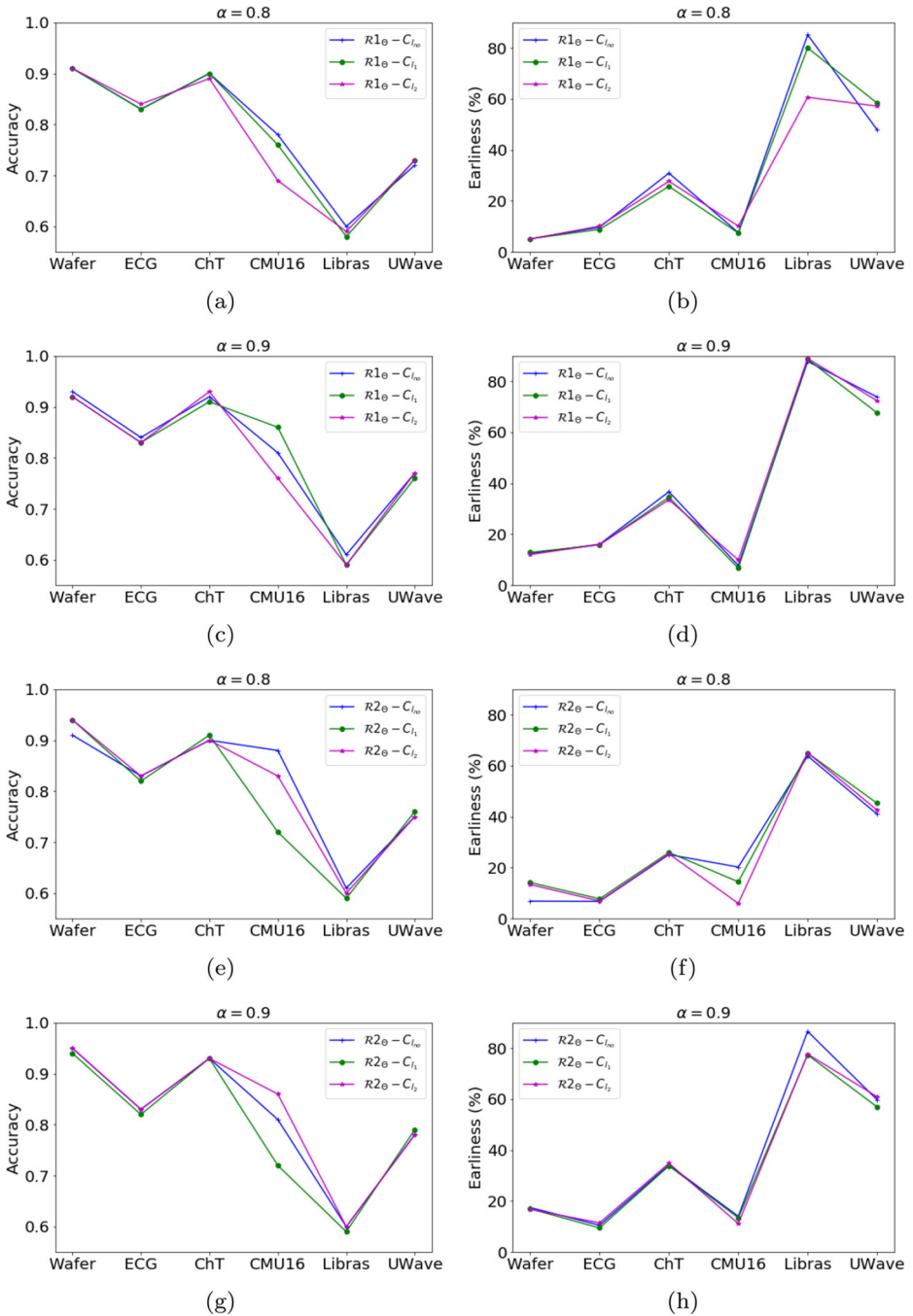


Fig. 10 Accuracy and earliness plot of ESRs for $\alpha \in \{0.8, 0.9\}$

the comparison based on *HM* metric clearly shows that the proposed model is the best performing one compared to MCFEC and MTSECP. It is also observed that MCFEC beats the

Table 2 Comparison of proposed models with other methods

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

MTSECP with 12% and 24% margin on *ECG* and *Wafer* datasets respectively. Whereas, the proposed model scores *HM* with high marginality about 11%, 10% and 36% respectively on *ECG*, *Wafer* and *ChT* datasets, 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 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 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 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 Fig. 7. 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.

6.4 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 11a and b illustrate the accuracy and earliness of various datasets. GP-Full indicates the GP classifier which is trained using full-length TS as per the conventional classification approach. Figure 11 shows that $\mathcal{R}1_{\Theta-C_{l_1}}$, $\mathcal{R}1_{\Theta-C_{l_2}}$, $\mathcal{R}2_{\Theta-C_{l_1}}$, and $\mathcal{R}2_{\Theta-C_{l_2}}$, have achieved decent accuracies over all the datasets as compared to GP-full by utilizing approximately 37% of full-length MTS. $\mathcal{R}2_{\Theta-C_{l_2}}$ 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

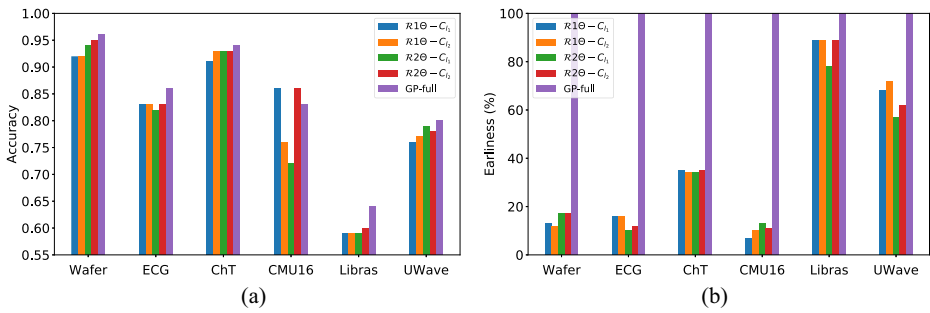


Fig. 11 Comparison of the proposed model with GP-full by considering $\alpha = 0.9$

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

Table 3 Accuracy and earliness of proposed model over six MTS datasets

	α	Wafer		ECG		ChT		CMU16		Libras		UWave	
		Acc	Ear	Acc	Ear	Acc	Ear	Acc	Ear	Acc	Ear	Acc	Ear
$\mathcal{R}_{1\Theta-C_{l_1}}$	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
$\mathcal{R}_{1\Theta-C_{l_2}}$	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
$\mathcal{R}_{2\Theta-C_{l_1}}$	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
$\mathcal{R}_{2\Theta-C_{l_2}}$	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
$\mathcal{R}_{1\Theta-C_{no}}$	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
$\mathcal{R}_{2\Theta-C_{no}}$	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

traditional TS 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 Fig. 7, 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.

7 Conclusion

In this paper, we proposed an optimization-based early classification model for MTS data by learning optimal Early Stopping Rules (ESRs). The ESRs continuously examine the output of probabilistic classifiers and predict the class label when enough data points become available in the incoming MTS. The ESRs have been learned through particle swarm optimization by minimizing the cost of accuracy and earliness. Besides, the proposed model has employed the Gaussian process probabilistic classifier at each variable of MTS 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 that can be chosen based on the requirement of the user. The proposed model has been evaluated on publically available datasets and has outperformed state-of-the-art methods by providing balancing trade-off between the objectives accuracy and earliness.

In future work, more complex weighted ESRs can be designed by assigning the higher weight to more informative components in MTS. Furthermore, the proposed model can be optimized for specific multimedia time-series applications such as early voice detection, gait recognition by customizing cost function, ESR, and other parameters of the proposed model.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Arzani MM, Fathy M, Azirani AA, Adeli E (2020) Skeleton-based structured early activity prediction. *Multimedia Tools and Applications*
2. Bagnall A, Lines J, Bostrom A, Large J, Keogh E (2016) The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min Knowl Disc* 31(3):606–660
3. Baydogan MG (2015) Multivariate time series classification datasets. <http://www.mustafabaydogan.com>
4. Bendtsen C (2012) pso: Particle Swarm Optimization. R package version 1.0.3
5. Bregón A, Simón MA, Rodríguez JJ, Alonso C, Pulido B, Moro I (2006) Early fault classification in dynamic systems using case-based reasoning. In: *Current topics in artificial intelligence*, Springer, Berlin pp 211–220
6. Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista G (2015) The ucr time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/
7. Dachraoui A, Bondu A, Cornuéjols A (2015) Early classification of time series as a non myopic sequential decision making problem. In: *Machine learning and knowledge discovery in databases*, Springer International Publishing, pp 433–447
8. Dua D, Graff C (2017) UCI machine learning repository. <http://archive.ics.uci.edu/ml>

9. Ghalwash MF, Radosavljevic V, Obradovic Z (2014) Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14, ACM Press
10. Ghalwash MF, Obradovic Z (2012) Early classification of multivariate temporal observations by extraction of interpretable shapelets. *BMC Bioinforma* 13(1):195
11. González CJA, Díez JJR (2004) boosting interval - based literals: variable length and early classification. In: Series in machine perception and artificial intelligence, WORLD SCIENTIFIC, pp 149–171
12. Gupta A, Gupta HP, Biswas B, Dutta T (2020) An early classification approach for multivariate time series of on-vehicle sensors in transportation. *IEEE Trans Intell Transp Syst*, pp 1–12
13. Hassan R, Cohanin B, de Weck O, Venter G (2005) A comparison of particle swarm optimization and the genetic algorithm. In: 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, American Institute of Aeronautics and Astronautics
14. He G, Duan Y, Peng R, Jing X, Qian T, Wang L (2015) Early classification on multivariate time series. *Neurocomputing* 149:777–787
15. He G, Zhao W, Xia X, Peng R, Wu X (2018) An ensemble of shapelet-based classifiers on inter-class and intra-class imbalanced multivariate time series at the early stage. *Soft Comput* 23(15):6097–6114
16. He G, Zhao W, Xia X (2019) Confidence-based early classification of multivariate time series with multiple interpretable rules. *Pattern Analysis and Applications*
17. Hoai M, la Torre FD (2013) Max-margin early event detectors. *Int J Comput Vis* 107(2):191–202
18. Hsu EY, Liu CL, Tseng VS (2019) Multivariate time series early classification with interpretability using deep learning and attention mechanism. In: Advances in knowledge discovery and data mining, Springer International Publishing, pp 541–553
19. Kate RJ (2015) Using dynamic time warping distances as features for improved time series classification. *Data Min Knowl Disc* 30(2):283–312
20. Lama N, Mark G (2016) vbmp: Variational Bayesian Multinomial Probit Regression, R package version 1.42.0
21. Li S, Li K, Fu Y (2018) Early recognition of 3d human actions. *ACM Transactions on Multimedia Computing, Communications, and Applications* 14(1s):1–21
22. Lin YF, Chen HH, Tseng VS, Pei J (2015) Reliable early classification on multivariate time series with numerical and categorical attributes. In: Advances in knowledge discovery and data mining, Springer International Publishing, pp 199–211
23. Lv J, Hu X, Li L, Li P (2019) An effective confidence-based early classification of time series. *IEEE Access* 7:96,113–96,124
24. Ma C, Weng X, Shan Z (2017) Early classification of multivariate time series based on piecewise aggregate approximation. In: Health information science, Springer International Publishing, pp 81–88
25. Mori U, Mendiburu A, Keogh E, Lozano JA (2016) Reliable early classification of time series based on discriminating the classes over time. *Data Min Knowl Disc* 31(1):233–263
26. Mori U, Mendiburu A, Dasgupta S, Lozano JA (2018) Early classification of time series by simultaneously optimizing the accuracy and earliness. *IEEE Transactions on Neural Networks and Learning Systems* 29(10):4569–4578
27. Ng AY (2004) Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In: Twenty-first international conference on Machine learning - ICML'04, ACM Press
28. Olszewski RT (2001) Generalized feature extraction for structural pattern recognition in time-series data. Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE
29. Parsopoulos K, Vrahatis M (2002) Recent approaches to global optimization problems through particle swarm optimization. *Nat Comput* 1(2):235–306
30. Richhariya B, Tanveer M (2018) EEG Signal classification using universum support vector machine. *Expert Syst Appl* 106:169–182
31. Rußwurm M, Tavenard R, Lefèvre S, Körner M (2019) Early classification for agricultural monitoring from satellite time series 1908.10283v1
32. Santos T, Kern R (2016) A literature survey of early time series classification and deep learning. In: Sami@ iknow
33. Sharma A, Singh SK (2019) Early classification of time series based on uncertainty measure. In: 2019 IEEE Conference on Information and Communication Technology, IEET
34. Sharma A, Singh SK (2020) A novel approach for early malware detection. *Transactions on Emerging Telecommunications Technologies* <https://doi.org/10.1002/ett.3968>
35. Tavenard R, Malinowski S (2016) Cost-aware early classification of time series. In: Machine learning and knowledge discovery in databases, Springer International Publishing, pp 632–647
36. Williams CK, Rasmussen CE (2006) Gaussian processes for machine learning 3. MIT Press, Cambridge
37. Xing Z, Pei J, Yu PS (2011) Early classification on time series. *Knowl Inf Syst* 31(1):105–127

38. Xing Z, Pei J, Yu PS, Wang K (2011) Extracting interpretable features for early classification on time series. In: Proceedings of the 2011 SIAM international conference on data mining, SIAM, pp 247–258
39. Yao L, Li Y, Li Y, Zhang H, Huai M, Gao J, Zhang A (2019) DTEC: Distance Transformation based early time series classification. In: Proceedings of the 2019 SIAM international conference on data mining, Society for Industrial and Applied Mathematics, pp 486–494
40. Zhao L, Liang H, Yu D, Wang X, Zhao G (2019) Asynchronous Multivariate time series early prediction for ICU transfer. In: Proceedings of the 2019 International Conference on Intelligent Medicine and Health - ICIMH 2019, ACM Press

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.