Early Classification of Time Series by Simultaneously Optimizing the Accuracy and Earliness

Usue Mori, Alexander Mendiburu, Sanjoy Dasgupta, and Jose A. Lozano, Member, IEEE

Abstract—The problem of early classification of time series appears naturally in contexts where the data, of temporal nature, are collected over time, and early class predictions are interesting or even required. The objective is to classify the incoming sequence as soon as possible, while maintaining suitable levels of accuracy in the predictions. Thus, we can say that the problem of early classification consists of optimizing two objectives simultaneously: accuracy and earliness. In this context, we present a method for early classification based on combining a set of probabilistic classifiers together with a stopping rule (SR). This SR will act as a trigger and will tell us when to output a prediction or when to wait for more data, and its main novelty lies in the fact that it is built by explicitly optimizing a cost function based on accuracy and earliness. We have selected a large set of benchmark data sets and four other state-of-the-art early classification methods, and we have evaluated and compared our framework obtaining superior results in terms of both earliness and accuracy.

Index Terms—Classification, cost-optimization, early classification, time series.

I. INTRODUCTION

TIME series classification [1], [2] is a supervised learning task that deals with predicting the class labels of time series as accurately as possible by using a training set of completely labeled full-length time series. An example application of this task is trying to identify which household device is working at a given time by using the electricity usage profiles [3]. Also, Kadous and Sammut [4] use electrocardiography (ECG) data to predict whether a patient has heart disease or not and also to recognize sign language signs.

The problem of early classification appears when the unlabeled time series are collected over time, and it is desirable to obtain the class label predictions as early as possible. For example, Evans et al. [10] show that the monitoring of patients and early identification of physiologic deterioration can be used to raise alerts and prevent crises in hospitalized patients. Also, Ghalwash et al. [11] mention early stock crisis identification; Bregón et al. [12] apply early classification to classify different types of faults in a simulated industrial plant; Hatami and Chira [13] attempt to classify a set of different odors as early as possible by using odor signals obtained from a set of sensors with the aim of identifying chemical leaks. Finally, in [14], an early classification approach is applied to detect and identify bird songs as early as possible, with the objective of saving memory and battery life of the recording devices.

It is obvious that the earliness of the predictions has an influence on their accuracy but, can this influence be quantified or modeled? Can we build models that are able to classify time series as early as possible while maintaining a suitable level of accuracy? This is exactly the aim of early classification, finding a tradeoff between two conflicting objectives: the accuracy of the predictions and their earliness [9]. However, the degree of conflict between these two objectives changes drastically from database to database, and as more data becomes available, the accuracy evolves differently in each database [14]. So, the problem of early classification can be described as an optimization problem in which two conflicting objectives must be optimized at the same time.

In this paper, we propose an early classification framework, departing from our preliminary work in [15], which is based on combining a set of probabilistic classifiers and an optimized...
stopping rule (SR). In the previous paper, we presented an initial version of the method, considering only a basic SR and only one specific cost function (CF). In this paper, we propose several different SRs and CFs, improve on the definition and design of the method, extend our experimentation to analyze the characteristics of our methodology, including the influence of the parameters in detail, and include a new state-of-the-art method in the comparative analysis.

The rest of this paper is organized as follows. In Section II, we present the problem of early classification and summarize the previous work on the topic. Section III presents the main contribution of this paper. In Section IV, the experimentation and validation of this method is carried out, and in Section V, we analyze the results in detail. Finally, in Section VI, we summarize the main conclusions and propose some future research directions.

II. EARLY CLASSIFICATION OF TIME SERIES

A time series is an ordered sequence of pairs of finite length \( L \) [9]

\[
TS = \{(t_1, x_1), (t_2, x_2), \ldots, (t_L, x_L)\}.
\] (1)

Note that timestamps \( t_i \) take positive and ascending real values, so we are dealing with an ordered sequence. Indeed, although in the most common examples the \( t_i \) values refer to temporal references (timestamps), other types of orderings can also be defined [1]. The values of the time series \( (t_i, x_i) \) are usually real numbers.\(^1\) Finally, a database of time series is a set of time series with no order, which can all be of the same length or can have different lengths.\(^2\)

Considering these definitions, time series classification is formally defined as a supervised data mining task in which, given a training set of complete time series and their respective class labels \( X = \{(TS_1, CL_1), (TS_2, CL_2), \ldots, (TS_n, CL_n)\}\), the aim is to build a model that is able to predict the class label of new unlabeled time series as accurately as possible [1], [2], [9], [14].

As an extension of time series classification, early classification of time series also departs from a completely labeled training set of time series. However, this problem appears when the new unlabeled time series are collected over time. In these contexts, early class predictions are sometimes important, for example, when collecting the data is expensive or due to the consequences associated with making late decisions [16]. However, maintaining a suitable level of accuracy is also usually an important requirement. To sum up, the key to early classification is not just to maximize accuracy, but rather to find a tradeoff between two conflicting objectives: accuracy and earliness.

This problem has been compared to other classical problems in machine learning, such as optimal stopping, feature selection, learning with incomplete data, and so on [16]. However, early classification has its own peculiarities, such as the temporal correlation in the data, so its appearance is fairly recent in the literature. Xing et al. [9] formally defined the problem of early classification for the first time and proposed a method named after the problem itself: early classification on time series (ECTS). This paper analyzes the stability of the nearest neighbor relationships in the training set over time. Based on this information, the method selects the training time series that can reliably be used at each timestamp within a 1NN classifier.

Another intuitive approach is to simply learn a model for each early timestamp and design different mechanisms to decide which predictions are reliable and which are not. Different types of models and reliability conditions result in a wide variety of early classification methods such as those proposed in [13], [16], and [17].

A new method, denominated ECDIRE [14], and based on controlling class discriminativeness and reliability of predictions over time, lies in between these two methods. The main objective of this method is to analyze the evolution of the accuracy of a set of probabilistic classifiers over time, with the aim of identifying the timestamps from whence it is safe to make predictions. Predictions will only be made in these timestamps or later and, so, a large number of predictions are avoided. As with the second type of method introduced in this section, ECDIRE also incorporates a reliability condition that must also be met, and is useful to discard outlier series that do not belong to any class.

Finally, a completely different strategy can be found in [11], [18], and [19], where the authors propose methods based on shapelets [5], which refer to subsequences of time series that can be used to discriminate a given class from the others. In the context of early classification, the trick relies on finding a library of shapelets that is useful to discriminate the classes as early as possible. The baseline of this type of early classification methodologies is presented in [18] and is denominated early distinctive shapelet classification (EDSC).

Note that, even if the problem of early classification is clearly an optimization problem, neither of the approaches introduced above deal with the problem of early classification from a cost minimization point of view. Only one recent work [20] deals with the problem from this perspective. These authors propose a complex meta-algorithm based on calculating the expectation of the cost of misclassification at time \( t \), together with the cost of making the decision at time \( t \). Unfortunately, this method has only been validated on one benchmark time series, it has not been compared with other state-of-the-art methods, and its code is not available for comparison.

In this paper, we propose an early classification method based on a very intuitive idea: we will combine a set of probabilistic classifiers and an SR which, when optimized by a suitable CF, will act as a reliability test. The novelty of the approach lies in the construction of the SR, which is based on an optimization process that will aim to find a tradeoff between earliness and accuracy.

As it will be seen, this approach is easy to understand for nonexpert users, because it only implies using a basic rule.
However, it shows improved performance over other much more complex approaches in the literature, and it is capable of adapting its behavior to the different shapes that the evolution of the accuracy over time can take, and to the different degrees of conflict between the objectives.

III. EARLY CLASSIFICATION OF TIME SERIES BY MINIMIZING A COST FUNCTION

In this section, we present our early classification method. The proposed framework will consist of a learning phase and a prediction phase, which will be explained in detail in the following sections.

A. Learning Phase

The goal of this phase is to train a model that provides early and accurate class predictions for new unlabeled time series. For this purpose, we will use a training set \( X = \{\{TS_1, CL_1\}, \{TS_2, CL_2\}, \ldots, \{TS_n, CL_n\}\} \) of labeled full-length time series of finite length, and we will proceed as follows.

Step 1 (Learn Probabilistic Classifiers): In the first step, we will train two sets of probabilistic classifiers.

1) On the one hand, we will train a set of classifiers \( \{h_l\}_{l=1}^L \) for all timestamps \( t \in \{1, 2, \ldots, L\} \), or for a user-defined subset of timestamps. In our experimentation, we use many different databases, of which we have no specific domain knowledge, \textit{a priori}. As such, based on a percentage of the series lengths, we have selected a sequence of equidistant time points, in which we carry out the classification. Nevertheless, the user could choose any other subset of timestamps, based on domain knowledge or other information of the shape of the series or even by using specific time series sampling methods. This is especially important if the series are of different or unknown lengths or if they are unevenly sampled.

To train these classifiers, we will use the whole training set \( X \). Each classifier \( h_l \) will receive the first \( t \) points of a series, and will output the posterior probabilities for each class at that time. These classifiers will be used to obtain the posterior class membership probabilities for the new unlabeled (test) time series at each time \( t \). We can see an illustrative example of the construction of the \( h_l \) classifiers in Fig. 1, for a training set of five time series.

2) On the other hand, we will build another set of classifiers \( \{h_{l,t}\}_{l=1}^L \), similar to \( h_l \) but trained using a sort of fivefold cross-validation scheme. We will use these classifiers to obtain the posterior probabilities of the training examples, but with the intention of avoiding the overfitting phenomenon. We can see an illustrative example of the construction of the \( h_{l,t} \) classifiers in Fig. 2, for a training set of five time series. Based on this construction, to obtain the posterior probabilities of a given training series \( TS \) at time \( t \), we will use the \( h_{l,t} \) classifier, which has been trained without using series \( TS \).

Step 2 (Definition of SRs): In the second step, we define the SRs and the associated CFs. Specifically, we propose two different SRs, of different characteristics, whose performance will be analyzed in the experimental section.

As a first simple SR, we analyze a basic SR (\( SR_1 \)), based on intuition and defined by the following linear rule, that is proposed previously by [15]:

\[
SR_1(p^t, t) = \begin{cases} 
0 & \text{if } \gamma_1 p_{1,k}^t + \gamma_2 (p_{1,k}^t - p_{2,k}^t) + \gamma_3 \frac{L}{T} \leq 0 \\
1 & \text{otherwise}
\end{cases}
\]  

where \( p^t = (p_{1,1}^t, p_{1,2}^t, \ldots, p_{1,k}^t) \) is the vector of posterior probabilities for the \( k \) possible classes issued by the corresponding \( h_l \) for a given time series, \( p_{1,k}^t \) and \( p_{2,k}^t \) are the first and second largest posterior probability values obtained at time \( t \), and \( \gamma = (\gamma_1, \gamma_2, \gamma_3) \) is a vector of parameters that takes real values between \(-1\) and \(1\).

The interpretation of this SR is the following: if the rule outputs a value of \(1\), we conclude that the prediction is reliable enough, and thus, the class corresponding to the maximum posterior probability value is provided. On the contrary, a value of \(0\) indicates that the prediction is not yet reliable, and so, we should wait until a larger part of the time series is available. If the entire time series is available and the SR has not triggered, the class prediction obtained at \( t = L \) is used.

As indicated above, the shape of the SR has been chosen based on intuition. We assume that the value of \( p_{1,k} \) and the difference between the two largest posterior probabilities are indicators of the goodness and reliability of the prediction obtained. As such, we include these two terms in the SR. Furthermore, the time in which the prediction is made can also...
be an influencing factor, so we also include this parameter in the SR. Note that in posterior steps, the \(\gamma_i\) parameters in this SR will be optimized using a suitable CF and so, each of the components of the SR will be given more or less relevance, depending on their corresponding \(\gamma_i\) value.

As a second and novel SR, we avoid defining the shape of the SR a priori and introduce all the posterior probabilities (SR2_\(\gamma\))

\[
SR_2(\mathbf{p}_t, t) = \begin{cases} 
0 & \text{if } \gamma_1 p_{t,1}^1 + \ldots + \gamma_k p_{t,k}^1 + \gamma_{k+1} 
\leq 0 \\
1 & \text{otherwise.}
\end{cases}
\]

(3)

**Step 3 (Definition of the CF):** Recall that the final aim is to find an optimized SR, which takes earliness and accuracy into account. For this, departing from the general SRs defined in the previous step, we will try to find the \(\gamma\) parameters that minimize a given CF, based on accuracy and earliness. In this section, we will propose various CFs of different characteristics, which will be compared later in the experimental section.

To begin with, we propose the following basic CF proposed initially in [15]:

\[
\text{CF1}(X, SR_\gamma) = \sum_{x \in X} \text{CF1}(x, SR_\gamma) \\
= \sum_{x \in X} (\alpha C_a(x, SR_\gamma) + (1 - \alpha) C_e(x, SR_\gamma))
\]

(4)

where \(\alpha \in [0, 1]\) is a user-defined parameter that represents the weight associated with each of the objectives, and \(C_a\) and \(C_e\) are the cost of accuracy and earliness, and will be presented at the end of this section.

Additionally, when we are using the SR2, we would like our optimization process to select some of the posterior probabilities from the SR automatically, based on their relevance. This means that we want to reward simpler models that will have as many 0s in the \(\gamma\) vector as possible. In optimization, this is denominated regularizing the CF for sparsity in the parameter vector.

The most direct manner of rewarding this sparsity is introducing a nonconvex penalization term to the basic CF, based on the 0-norm of the chosen parameters. This term will penalize the vectors of parameters that have many nonzero components and will result in the following new CF (CF2):

\[
\text{CF2}(X, SR_\gamma) = \sum_{x \in X} \text{CF2}(x, SR_\gamma) \\
= \sum_{x \in X} (\alpha C_a(x, SR_\gamma) + (1 - \alpha) C_e(x, SR_\gamma)) - \lambda \lVert \gamma \rVert_0.
\]

(5)

Note that this penalization term is nonconvex, and its application results in NP-hard optimization problems. As such, in practice, it is common to use a relaxed but convex penalization term for sparsity, based on the \(\lVert \gamma \rVert_1\) norm. This is typically done in the basic LASSO method, where this penalization term is added to the classic least squares formulation for regression [21]. The introduction of this penalization term in the CF will result in the third CF that will be considered in this paper

\[
\text{CF3}(X, SR_\gamma) = \sum_{x \in X} \text{CF3}(x, SR_\gamma) \\
= \sum_{x \in X} (\alpha C_a(x, SR_\gamma) + (1 - \alpha) C_e(x, SR_\gamma)) - \lambda \lVert \gamma \rVert_1.
\]

(6)

The functions \(C_a\) and \(C_e\), which appear in all the CFs, can be defined in several ways. In this case

\[
C_e(x, SR_\gamma) = \frac{t^*_x}{L}
\]

(7)

t^*_x being the earliest timestamp in which \(s(t)\) outputs a value of 1 (halt) for series \(x\), and \(L\) its true class value. \(I(\cdot)\) takes a value of 1 if the condition is true, and 0 otherwise. In essence, this is the classical 0-1 CF, based on comparing the true label with the label obtained at time \(t^*_x\).

Based on the two SRs and the three CFs, we obtain the following combinations: SR1-CF1, SR2-CF1, SR2-CF2, and SR2-CF3.

**Step 4 (Optimization Process):** The \(\gamma\) vector that minimizes this CF can be found by using several different optimization algorithms. Of course, we must take into account the nature of the defined CF. Specifically, the nonconvexity of the CF and its lack of differentiability, among other features, will be determinant when choosing the optimization algorithm. The specific algorithms chosen in this paper will be introduced in the experimental section, but note that other optimization algorithms which do not require convexity or differentiability of the CF could be used without loss of generality.

In order to apply any optimization algorithm, we have to be able to, at least, evaluate the objective function at different points. Note that the objective function is defined as the cost associated with different \(\gamma\) values. Given a training set of time series \((X)\) for which the true class value is known, in Fig. 3, we can see an example of how the cost associated with a given training example \((x)\) would be computed for a certain SR and a specific parameter vector \((\gamma)\). Consequently, the cost of the whole training set is calculated by summing the costs of all its conforming time series.

As can be seen, starting from timestamp \(t = 1\), the posterior probabilities for the training examples are obtained by using the corresponding \(h_t^i\) probabilistic classifiers, as explained previously. These posterior probabilities are introduced into the SR, which is completely defined by the chosen \(\gamma\) vector.
Fig. 4. Early classification of a test example $x_{\text{test}}$.

If the SR returns a value of 1, the process terminates, and the current class prediction ($\hat{C}_{L_i} = \arg\max_{i=1,...,k} \{p_{x_i}^{t^*}\}$) and timestamp ($t^*$) will be used directly to calculate the cost. If the SR outputs a 0, then we must add the next data point to the time series and repeat the process.

B. Prediction Step

Finally, once the framework has been trained, we can use it to predict the class labels of new time series. As can be seen in Fig. 4, each time a new data point of the new time series becomes available, we will introduce the truncated series into the corresponding $h_t$ classifier specific for that $t$ and trained with all the training set $X$ as explained before. This classifier, in turn, will output a vector of posterior probabilities for that time series and that specific time stamp $t$. Next, this probability vector will be introduced into the optimized SR, together with the current time stamp. The SR will provide an answer of 0 or 1, and based on this, we will halt and provide a class prediction, or, on the contrary, wait until more data are available.

Note that, for each database, the accuracy evolves differently as more data become available. As shown in [14], in some databases, the accuracy is strictly increasing. In others, in contrast, the accuracy becomes stable after some point, and in some databases, the accuracy even drops after some point due to the noise that the additional data incorporates. The proposed method has been designed to be very flexible and does not require the stability or convergence of the SRs, as do other methods in the literature such as [9], [14]. This enables a better adjustment to the different forms that the accuracy can take over time and allows the method to profit more from early class predictions.

IV. EXPERIMENTAL SETUP

In this section, we detail the setting of the experiments carried out to evaluate our method.3

A. Data

To evaluate our proposal, we have considered the 45 databases available at the time of experimentation from the University of California, Riverside (UCR), time series database archive [22]. This archive collects the majority of the publicly available synthetic and real time series databases designed for clustering and classification purposes, and is the classical benchmark used for evaluation in the study area.

3The complete code of the proposed methodology is available in http://www.sc.ehu.es/ccwymbayes/members/umori/EarlyClassification/EarlyClassification.html

B. Parameter Selection

As aforementioned, in this section, the $h_t$ and $h'_t$ classifiers can be trained in every timestamp, $t = \{1, \ldots, L\}$, but a smaller set of timestamps can also be chosen. In this case, due to the large number of databases considered and because we want to control the computational cost of the experimentation, we build the classifiers only with a granularity of 5% of the length of time series. However, some additional experiments that consider other granularities and analyze the effect of this parameter are provided in Section V-B.

To build these two sets of probabilistic classifiers, we have used Gaussian process (GP) classifiers [23], because they have shown good performance in similar frameworks [14], [15]. Nevertheless, any other probabilistic model could be used equivalently within our framework. Indeed, in order to obtain some additional insights, and to analyze the effect of using another alternative probabilistic classifier, in Section V-D, we show the results of some experiments using support vector machines (SVM) combined with Platt’s scaling [24].

The GPs are implemented using an extension of the vbmp package of R [25] previously used in [14]. Similar to this paper, the parameters for the vbmp function have all been set to their default values except for the convergence threshold (set to $10^{-6}$), and the maximum iteration number (set to 24). Finally, after preliminary experiments, the inner product kernel has been chosen as the covariance function. On the contrary, the SVM classifiers have been trained using the e1071 R package [26], using the Gaussian kernel and leaving the remaining parameters in their default values.

In the previous studies, the use of specific distance measures for time series, such as dynamic time warping or edit distances has shown to improve the results of classification algorithms when dealing with time series classification [27]. As such, following the example of [14] and [28], we enable the use of different distance measures within the probabilistic classifiers, by using a suitable distance matrix as input to the classifiers, instead of the raw time series. This distance matrix will save the pairwise distances between all the series in the training set, and can be built using any distance measure of choice which allows us to deal with discrete series and series of different lengths, simply by choosing a suitable distance matrix. However, since the objective is not to evaluate the performance of different distance measures, in this experimentation, we always use the standard Euclidean distance.

Regarding the $\alpha$ parameter, which appears in the CFs defined in Section III-A, we analyze four weight values ($\alpha \in \{0.6, 0.7, 0.8, 0.9\}$). The reason why we only choose $\alpha$ values higher that 0.5 is that the other early classification methods that we will use for comparison (see the next section) aim at obtaining the same accuracy that is obtained when the full series is available. In this context, we will also lean toward this objective. However, in Section V-A, we will analyze the effect of modifying this parameter.

As mentioned previously, different optimization algorithms can be used within the presented framework, always taking into account the properties of the defined CF. In our case, since we cannot calculate an analytic expression of the gradient of the CF, and we do not always have information about its
convexity, we have chosen to use genetic algorithms (GA) [29]. These algorithms make few assumptions about the search space and are suitable to solve many kinds of complex optimization problems. As such, we can also use this optimization algorithm when the shape of the CF is unknown or nonconvex and, so, the optimization problem cannot be solved efficiently by standard algorithms. We implement the genetic algorithm using the 	extit{ga} function of the 	extit{GA} package written in R [30] using the default parameter values of this function. The population size is 50, and the initial population is chosen uniformly at random. With regard to the genetic operators, at each iteration, a whole new population is created by first applying selection, carried out by using fitness proportional selection with linear scaling [31]. Then, using the local arithmetic crossover operator, these recombined solutions are recombined directly to enable reproducibility and comparison. Furthermore, the two objectives of the early classification problem are evaluated separately based on the previously defined cost function. We compare the methods with the four state-of-the-art early classification methods [4, 5, 6 A summary of the methods chosen and the parameter configurations experimented with are summarized in Table I. Note that this selection has been made based on the experimentation carried out by the corresponding authors.

Note that, additionally, in the Rel.Class. method, the local discriminative Gaussian dimensionality reduction has been enabled because it reduces computational costs and can reduce noise and yield higher accuracy values [16], and the joint Gaussian estimation method has been chosen because it is more efficient and obtains similar results to those obtained by the other estimation methods considered.

D. Evaluation Method

The databases from the UCR archive are provided with prespecified training and testing sets, which have been used directly to enable reproducibility and comparison. Furthermore, the two objectives of the early classification problem are evaluated separately based on the previously defined cost of accuracy ($C_a$) and cost of earliness ($C_e$)

\[
\text{Accuracy} = \frac{1}{|X_{\text{test}}|} \times \sum_{x \in X_{\text{test}}} \mathbb{I}(\text{argmax}_{i=1,\ldots,k}(p_{x}^{i}) = \text{CL}_i) \\
\text{Earliness} = \frac{1}{|X_{\text{test}}|} \sum_{x \in X_{\text{test}}} \frac{t_x^e}{L} \cdot 100\%. \tag{10}
\]

In addition to measuring and comparing these two objectives separately, we must consider the multiobjectiveness of the problem. As such, we use the Pareto optimality criterion as in [14], which states that a solution dominates another if it obtains better results in at least one of the objectives while not degrading any of the others. We compare the methods pairwise and calculate the number of times in which our method dominates the other and vice versa.

Based on the domination counts, we also calculate two summary indices: \textit{Summary 1} is calculated by assigning one point for each database in which our method wins and subtracting one point for each database in which our method loses; \textit{Summary 2} is calculated by assigning one point for winning, half point for each draw, and 0 points for losing.

V. RESULTS

In Table II, we show the domination counts for each configuration of our method (SR1-CF1, SR2-CF1, SR2-CF2, and SR2-CF3) compared to the other state-of-the-art

| Method  | Variants          | Parameter name | Values       |
|---------|-------------------|----------------|--------------|
| ECTS [9]| -Strict           | Minimum support| 0, 0.05, 0.1, 0.2, 0.4, 0.8 |
|         | -Loose            |                |              |
| EDSC [18]| Chebyshev Inequality version | Chebyshev bound | 2.5, 3, 3.5 |
| RelClass [16]| -Naive Gaussian Quadratic set | Reliability threshold | 0.001, 0.1, 0.5, 0.9 |
|         | -Gaussian Naive Bayes box | $\tau$        |              |
| ECDIRE  |                  | acc_perc       | 100%         |

\textit{ECTS} [9] and \textit{EDSC} [18]: \texttt{http://zhengzhengxing.blogspot.com.es/p/research.html}

\textit{Rel.Class.} [16]: \texttt{http://www.mayagupta.org/publications/Early_Classification_For_Web.zip}

\textit{ECDIRE} [14]: \texttt{http://www.sc.chu.es/ccwbayes/members/umori/ECDIRE/ECDIRE.html}
TABLE II
DOMINATION COUNTS FOR OUR METHOD WITH GPs IN COMPARISON TO
ECDIRE, ECTS, EDSC, AND REL. CLASS. THE FIRST NUMBER COR-
RESPONDS TO THE NUMBER OF TIMES OUR METHOD DOMINATES
THE OTHER METHOD, THE SECOND NUMBER REFERS TO THE
DRAWS, AND THE THIRD NUMBER COUNTS THE TIMES THE
COMPARISON METHOD DOMINATES OUR METHOD

| α = 0.6 | ECDIRE | Rel. Class. | ECTS | EDSC | Summary1 | Summary2 |
|---------|--------|------------|------|------|----------|----------|
| SR1-CF1 | 15/30/0| 11/33/1    | 21/23/1 | 29/16/0 | 18.50  | 31.75 |
| SR2-CF1 | 18/27/0| 7/37/1     | 18/26/1 | 29/16/0 | 17.50  | 31.25 |
| SR2-CF2 | 16/29/0| 9/35/1     | 18/26/1 | 29/16/0 | 17.50  | 31.25 |
| SR2-CF3 | 18/27/0| 7/37/1     | 18/26/1 | 29/16/0 | 17.50  | 31.25 |
|          |        |            |       |       | 17.75  | 31.38 |

| α = 0.7 | ECDIRE | Rel. Class. | ECTS | EDSC | Summary1 | Summary2 |
|---------|--------|------------|------|------|----------|----------|
| SR1-CF1 | 18/27/0| 12/31/2    | 24/20/1 | 31/13/1 | 20.25  | 32.63 |
| SR2-CF1 | 20/25/0| 14/29/2    | 23/21/1 | 30/14/1 | 20.75  | 32.88 |
| SR2-CF2 | 21/24/0| 15/28/2    | 25/19/1 | 30/14/1 | 21.75  | 33.38 |
| SR2-CF3 | 20/25/0| 14/29/2    | 23/21/1 | 30/14/1 | 20.75  | 32.88 |
|          |        |            |       |       | 20.88  | 32.94 |

| α = 0.8 | ECDIRE | Rel. Class. | ECTS | EDSC | Summary1 | Summary2 |
|---------|--------|------------|------|------|----------|----------|
| SR1-CF1 | 28/17/0| 22/21/2    | 28/15/2 | 31/14/0 | 26.25  | 35.63 |
| SR2-CF1 | 25/20/0| 20/23/2    | 29/15/1 | 30/15/0 | 25.25  | 35.13 |
| SR2-CF2 | 27/18/0| 22/21/2    | 29/15/1 | 31/14/0 | 26.50  | 35.75 |
| SR2-CF3 | 25/20/0| 20/23/2    | 29/15/1 | 30/15/0 | 25.25  | 35.13 |
|          |        |            |       |       | 25.81  | 35.41 |

| α = 0.9 | ECDIRE | Rel. Class. | ECTS | EDSC | Summary1 | Summary2 |
|---------|--------|------------|------|------|----------|----------|
| SR1-CF1 | 28/17/0| 23/19/3    | 29/15/1 | 28/17/0 | 26.00  | 35.50 |
| SR2-CF1 | 24/20/1| 23/19/3    | 32/11/2 | 28/17/0 | 25.25  | 35.13 |
| SR2-CF2 | 22/22/1| 24/19/2    | 32/11/2 | 28/17/0 | 25.25  | 35.13 |
| SR2-CF3 | 24/20/1| 23/19/3    | 32/11/2 | 28/17/0 | 25.25  | 35.13 |
|          |        |            |       |       | 25.44  | 35.22 |

Fig. 5. Evolution of accuracy and earliness according to α.

methods (ECDIRE, Rel.Class., ECTS, and EDSC). We also show the scores Summary 1 and Summary 2, for each combination of our method and, also, average values for each α value.

Note that, for ECDIRE, Rel.Class., ECTS, and EDSC, from all the parameter combinations enumerated in Table I, which we have considered in the experimentation, we only show the results for the configuration that dominate our method the most times. If there are ties within a method, the configuration that is dominated a lower number of times by our method is chosen. Anyhow, the raw accuracy and earliness values obtained by all the methods and all the parameter configurations can be found in our website.

As we can see, all four proposed methods dominate the rest of the methods by a large margin, proving that our methodology is an effective way of performing early classification. Furthermore, if we analyze the statistical significance of these results using a signed permutation test [34] (corrected with the Holm–Bonferroni post-hoc test to control the family-wise error), all the p-values that we obtain except SR2-CF1 or SR2-CF3 with α = 0.6 compared to Rel.Class., indicate significant differences for a significance level of 0.05, supporting the previous statement further. The p-values issued from these statistical tests are available at http://www.sc.ehu.es/ccwbayes/members/umori/EarlyClassification/pvalues.pdf. As such, these results indicate that our method, by means of the posterior probabilities obtained from the classifiers and the optimization process, is able to analyze how the accuracy evolves as more data become available, and is capable of identifying when the best moment to stop and provide a prediction is.

Additionally, if we study the domination counts and also the values obtained by the two summary indexes (Summary1 and Summary2), we can see that, overall, α = 0.8 seems to achieve the most impressive differences when compared to the other state-of-the-art methods.

If we compare the different combinations of our method, combinations SR1-CF1 and SR2-CF2 obtain the highest values for Summary1 and Summary2, being the former the winner when α is 0.6 or 0.9 and the latter when it is 0.7 or 0.8.

After analyzing these general results, in the next sections, we will provide a more detailed analysis and discussion of the effect of modifying the different parameters of the framework: the α parameter, the sampling granularity of the series, the SR, the CF, and the underlying classifier.

A. Effect of Modifying α

Based on the formulation of the CFs, when we increase α, we expect the accuracy to increase whereas the earliness values should become worse. To show that this is indeed so, in Fig. 5, we plot the mean value (averaged over all databases) of the accuracy and earliness for different α values.

As we can see, the trend of the plots is clearly increasing, for all four proposed methods. In this sense, if accuracy
requirements were lower, we could use other smaller $\alpha$ values.

In any case, as commented previously, if we analyze the domination counts obtained by different $\alpha$ values in comparison to the rest of the state-of-the-art methods (see Table II), it seems that the best results are obtained by larger $\alpha$ values. This is somewhat expected, since all the rest of the methods aim at obtaining 100% accuracy. Specifically, if we consider the values of the performance scores $\text{Summary1}$ and $\text{Summary2}$, both considering each SR and CF combination separately, and also averaging over them, the best results are obtained by $\alpha = 0.8$. As such, from now on, we will only take into account this value of $\alpha$.

However, note that the selection of the most suitable $\alpha$ can be made following the other criteria, and it strongly depends on the database and also on the interest or requirements of the user in terms of accuracy and earliness.

### B. Effect of Modifying the Granularity of the Sampling

The sampling granularity of the time series, which has been set to 5%, is also a parameter of the model. As such, with the aim of analyzing to what extent the results are affected by this parameter, we have carried out some experiments based on the SR1-CF1 combination and $\alpha = 0.8$, but using different granularities. The accuracy and earliness summaries can be found in Fig. 6, and the domination counts when compared to the other state-of-the-art methods are shown in Table III.

![Fig. 6. Accuracy and earliness values for the considered databases, depending on the sampling granularity.](image)

### C. Effect of Modifying the Cost Function and the Stopping Rule

In order to analyze the effect of modifying the CF and SR, we select $\alpha = 0.8$ as explained previously, and in Table IV, we compare the results obtained by the four configurations of our method.

First, if we analyze the effect of the different SRs, we must emphasize that the simplest SR (SR1) obtains quite good results when comparing it with other state-of-the-art methods (see Table II). However, it performs a little bit worse than the other combinations of our own method. In Table IV, we can see that the SR2 rule obtains slightly higher domination counts than SR1, for all three CFs considered. Even if these differences are not statistically significant, it seems that using SR2 better adjusts to the characteristics of some databases and is thus a better choice, in general.

Second, if we analyze the different CFs by comparing the results obtained by SR2-CF1, SR2-CF2, and SR2-CF3, we conclude that applying regularization improves the results slightly. With $\alpha = 0.8$, SR2-CF2 obtains the best results when compared to other state-of-the-art algorithms (see Table II). This is because it obtains very good earliness values, while maintaining competitive results on accuracy. Contrarily, SR2-CF3 obtains results almost identical to those obtained by its unregularized version (SR2-CF1), but it improves on the earliness in two data sets. SR2-CF3 tends more toward accuracy than SR2-CF2, which results in higher domination counts for the former when we compare these two configurations among each other (see Table IV). However, SR2-CF2 seems to benefit more from the regularization than SR2-CF3.

### D. Effect of Modifying the Underlying Classifier

In order to compare the performance of GP and the SVM classifiers within our framework, we choose the most simple approach (SR1-CF1), and in Table V, we show the domination counts of our method, using SVMs as underlying classifiers.
instead of GPs. Based on these results, it is evident that the GP classifiers obtain much better results.

As expected, our framework is limited to the prediction ability of the chosen underlying classifiers, since they will set bounds to the maximum accuracy that can be obtained. In this case, the SVMs do not yield high accuracy results, and as such, when we increase the value of $\alpha$, the domination values remain quite low, especially when we compete with methods that obtain high accuracy values. However, note that even when using SVM classifiers, our method dominates the rest of the comparison methods more times that it is dominated in most cases, which proves the usefulness of our method.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an early classification framework based on combining a set of probabilistic classifiers and an SR, designed by minimizing the cost in earliness and accuracy. The method is conceptually simple and does not require complex parameter settings. Furthermore, it is one of the few approaches that tackle the problem of early classification from an optimization point of view.

We have experimented with different CFs conformed with these two objectives and also with different SRs by using 45 benchmark databases from the UCR archive. We have also compared our results to other early classification methods from the state-of-the-art, showing superior results in terms of earliness and accuracy.

As future work, we propose designing a more complex structure for the SR, in which a different SR is associated with each class or more common reliability measures are used. The idea is to capture the different behaviors of the classes, if they exist. Additionally, more complex methods, such as genetic programming, could be used to learn more suitable SRs in a more automatic manner.

Finally, the problem of early classification has two conflicting objectives. In this paper, the balance between these two objectives has been sought by means of the $\alpha$ parameter, and we have emphasized that the choice of a suitable $\alpha$ strongly depends on the database at hand, and also on the requirements and needs of the user. In this sense, providing strategies to optimize this parameter in some specific context could be an interesting future research line. Additionally, and in this same line, it could be interesting to approach this problem as a multiobjective optimization problem, in which no $\alpha$ parameter would be necessary.

ACKNOWLEDGMENT

The authors would like to thank the UCR archive for providing access to the data used in this paper.

REFERENCES

[1] Z. Xing, J. Pei, and E. Keogh, “A brief survey on sequence classification,” ACM SIGKDD Explorations Newsletter, vol. 12, no. 1, pp. 40–48, 2010.
[2] W. Pei, H. Dibeklioğlu, D. M. J. Tax, and L. van der Maaten, “Multivariate time-series classification using the hidden-unit logistic model,” IEEE Trans. Neural Netw. Learn. Syst., to be published.
[3] J. Lines, A. Bagnall, P. Caiger-Smith, and S. Anderson, “Classification of household devices by electricity usage profiles,” in Intelligent Data Engineering and Automated Learning—IDEAL, H. Yin, W. Wang, and V. Rayward-Smith, Eds. Berlin, Germany: Springer, 2011, pp. 403–412.
[4] M. W. Kadous and C. Sammut, “Classification of multivariate time series and structured data using constructive induction,” Mach. Learn., vol. 58, nos. 2–3, pp. 179–216, 2005.
[5] L. Ye and E. Keogh, “Time series shapelets: A novel technique that allows accurate, interpretable and fast classification,” Data Min. Knowl. Discov., vol. 22, nos. 1–2, pp. 149–182, 2011.
[6] C. Li, L. Khan, and B. Prabhakaran, “Feature selection for classification of variable length multivariate motions,” Knowl. Inf. Syst., vol. 10, no. 2, pp. 163–183, 2006.
[7] L. Wei and E. Keogh, “Semi-supervised time series classification,” in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2006, pp. 748–753.
[8] M. González, C. Bergmeir, I. Triguero, Y. Rodríguez, and J. M. Benítez, “On the stopping criteria for k-nearest neighbor in positive unlabeled time series classification problems,” Inf. Sci., vol. 328, pp. 42–59, Jan. 2016.
[9] Z. Xing, J. Pei, and P. S. Yu, “Early classification on time series,” Knowl. Inf. Syst., vol. 31, no. 1, pp. 105–127, Apr. 2011.
[10] S. Evans et al., “Automated detection of physiologic deterioration in hospitalized patients,” J. Am. Med. Inform. Assoc., vol. 22, no. 2, pp. 350–360, 2015.
[11] M. F. Ghulwash, V. Radosavljevic, and Z. Obradović, “Utilizing temporal patterns for estimating uncertainty in interpretable early decision making,” in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2014, pp. 402–411.
[12] A. Bregón, M. A. Simón, J. J. Rodríguez, C. Alonso, B. Pulido, and I. Moro, “Early fault classification in dynamic systems using case-based reasoning,” in Proc. 11th Spanish Assoc. Conf. Current Topics Artif. Intell. (CAEPIA), 2009, pp. 211–220.
[13] N. Hatami and C. Chira, “Classifiers with a reject option for early time-series classification,” in Proc. IEEE Symp. Comput. Intell. Ensemble Learn. (CIEL), Apr. 2013, pp. 9–16.
[14] U. Mori, A. Mendiburu, E. Keogh, and J. A. Lozano, “Reliable early classification of time series based on discriminating the classes over time,” Data Min. Knowl. Discov., vol. 31, no. 1, pp. 233–263, 2016.
[15] U. Mori, A. Mendiburu, S. Dasgupta, and J. A. Lozano, “Early classification of time series from a cost minimization point of view,” in Proc. Neural Inf. Process. Syst. Time Ser. Workshop (NIPS), 2015, pp. 1–5.
[16] N. Parrish, H. S. Anderson, M. R. Gupta, and D. Y. Hsiao, “Classifying with confidence from incomplete information,” J. Mach. Learn. Res., vol. 14, no. 1, pp. 3561–3589, 2013.
[17] M. F. Ghulwash, D. Ramljak, and Z. Obradović, “Early classification of multivariate time series using a hybrid HMM/SVM model,” in Proc. IEEE Int. Conf. Bioinform. Biomed., Oct. 2012, pp. 1–6.
[18] Z. Xing, J. Pei, P. S. Yu, and K. Wang, “Extracting interpretable features for early classification on time series,” in Proc. 11th SIAM Int. Conf. Data Mining, 2011, pp. 247–258.
[19] G. He, Y. Duan, R. Peng, X. Jing, T. Qian, and L. Wang, “Early classification of multivariate time series,” Neurocomputing, vol. 149, pp. 777–787, Feb. 2015.
[20] A. Dachraoui, A. Bondu, and A. Cornuéjols, “Early classification of time series as a non myopic sequential decision making problem,” in Proc. ECML PKDD, 2015, pp. 433–447.
[21] R. Tibshirani, “Regression shrinkage and selection via the lasso,” J. Roy. Stat. Soc. B, vol. 58, pp. 267–288, 1994.

### Table V

| Method | SVMs | Rel. Class. | ECTS | EDSC |
|--------|------|------------|------|------|
| α = 0.6 | 3/4/1 | 4/3/2 | 6/3/2 | 22/23/0 |
| α = 0.7 | 3/4/1 | 4/3/2 | 6/3/2 | 24/21/0 |
| α = 0.8 | 5/3/6 | 4/3/6 | 7/3/6 | 25/17/3 |
| α = 0.9 | 5/3/1/9 | 2/3/6/7 | 10/3/3/2 | 20/23/2 |
Usue Mori received the M.Sc. degree in mathematics and the Ph.D. degree in computer science from the University of the Basque Country UPV/EHU, Donostia, Spain, in 2010 and 2015, respectively.

Since 2015, she has been a Lecturer with the University of the Basque Country UPV/EHU. Her current research interests include clustering and classification of time series.

Alexander Mendiburu received the B.Sc. degree in computer science and the Ph.D. degree from the University of the Basque Country UPV/EHU, Donostia, Spain, in 1995 and 2006, respectively.

Since 1999, he has been a Lecturer with the Department of Computer Architecture and Technology, University of the Basque Country UPV/EHU. His current research interests include evolutionary computation, time series, probabilistic graphical models, and parallel computing.

Sanjoy Dasgupta received the Ph.D. degree from University of California at Berkeley, Berkeley, CA, USA, in 2000.

He spent two years at the AT&T Research Labs, Florham Park, NJ, USA. He is currently a Professor with the Department of Computer Science and Engineering, University of California at San Diego, San Diego, CA, USA. He has authored a textbook, “Algorithms” (with Christos Papadimitriou and Umesh Vazirani) that appeared in 2006. His current research interest includes algorithmic statistics, with a focus on unsupervised and minimally supervised learning.

Dr. Dasgupta was a Program Co-Chair of the Conference on Learning Theory in 2009 and of the International Conference on Machine Learning in 2013.

Jose A. Lozano (M’05) received the Ph.D. degree in computer science from the University of the Basque Country UPV/EHU, Donostia, Spain, in 1998.

Since 2008, he has been a Full Professor with the University of the Basque Country UPV/EHU, where he currently leads the Intelligent Systems Group. He is the co-author of more than 100 ISI journal publications and a co-editor of three books. His current research interests include machine learning, pattern analysis, evolutionary computation, data mining, metaheuristic algorithms, and real-world applications.

Dr. Lozano is an associate editor of the IEEE Transactions on Neural Networks and Learning Systems.