Early classification of time series using multi-objective optimization techniques

U Mori[a,*], A. Mendiburu[b], I.M . Miranda[c], J.A. Lozano[d]

[a] *Department of Applied Mathematics, Statistics and Operations Research, University of the Basque Country UPV/EHU, 48940 Leioa, Spain.*

[b] *Department of Computer Arquitecture and Technology, University of the Basque Country UPV/EHU, 20018 Donostia, Spain.*

[c] *Departamento de Economía de la Empresa (Área de Finanzas), Universidad Rey Juan Carlos, Paseo de los Artilleros, s/n, 28032 Madrid.*

[d] *Basque Center for Applied Mathematics, 48009 Bilbo, Spain, and also with the Department of Computer Science and Artificial Inteligence, University of the Basque Country UPV/EHU, 20018 Donostia, Spain*

Abstract

In early classification of time series the objective is to build models which are able to make class-predictions for time series as accurately and as early as possible, when only a part of the series is available. It is logical to think that accuracy and earliness are conflicting objectives, since the more we wait, more data points from the series are available, and it is easier to make accurate class-predictions. Considering this, the problem can be very naturally formulated as a multi-objective optimization problem, and solved as such. However, the solutions proposed in the literature up to now, reduce the problem into a single-objective problem by combining both objectives somehow. In this paper, we present a novel multi-objective formulation of the problem of early classification, and we design a solution using multi-objective optimization techniques. This method will provide a variety of solutions which find different trade-offs between both objectives, allowing the user to select the most suitable solution a-posteriori, depending on the accuracy and earliness requirements of the problem at hand. To prove the usefulness of our proposal, we carry out an extensive experimentation process using 45 benchmark databases and we present a case study in the financial domain.

[*] Corresponding author

Email address: usue.mori@ehu.eus (U Mori)

1. Introduction

Lately, the analysis and pattern extraction from temporally ordered data, has become one of the most popular areas in data mining, due mainly to the proliferation of databases containing this type of data. Time series, ordered and real valued sequences of finite length [36], are among the most typical type of temporal data, and over the years, researchers have tried to extend many classical data mining solutions and algorithms to databases in which the instances are time series [9]. The main characteristic of this type of data is the temporal correlation between the measurements [12].

In this paper, we will focus on a supervised learning problem denominated *early classification of time series*, a variant of the classic classification problem. This problem appears when the new instances which are to be classified are time series and they are collected over time. In this context, the main objective is to learn a classification model, using a training set of complete and labeled time series, which will be able to predict the classes of new unlabeled time series as accurately and also as early as possible, preferably before their full-length is available [27].

As can be seen, this problem can be quite naturally understood as a multi-objective problem, where the accuracy and earliness of the class-predictions must be optimized simultaneously. These two objectives are conflicting, but equally important and different trade-offs between them can be useful on each occasion: some users may be interested in very early predictions, even if they must sacrifice on accuracy, and other more conservative users, may be willing to wait more in order to ensure the accuracy of the predicted labels. As such, it makes sense to try to find a set of solutions which range over all the possible trade-offs instead of just finding one solution. Then, the user can choose the solution which is more suitable for the specific application at hand.

The early classification methods proposed in the literature can be divided into two main groups. The first group of methods is based on shapelets [38], which are subsections of the time series which are useful to discriminate between the classes. In early classification, the aim is to search for shapelets which appear early in time [14, 16, 37]. The second group combines a set of classifiers, built in different timestamps, with one or various conditions or trigger functions which will evaluate the reliability of the predictions at different timestamps and help us decide whether the obtained prediction must be considered or discarded [13, 15, 26, 27, 28, 36]. Most of these methods focus on obtaining the 100% of the accuracy that would be obtained if the full time series were available, but earlier in time. As such, they are not designed to treat the two objectives equally and they tend strongly towards one of the objectives: accuracy. Some of them such as [6, 26, 27, 28, 32], include some user-defined parameters which somehow enable modifying this stiff trade-off between the two objectives, but these parameters are usually difficult to tune in advance, and, additionally, in order to obtain solutions with varying trade-offs, we must execute the algorithms more than once. Indeed, to the best of our knowledge, all the proposed solutions reduce the problem of early classification to a single-objective problem by combining the two objectives in some manner.

In this context, and with the aim of dealing with these drawbacks, we formulate the problem of early classification of time series as a multi-objective problem for the first time and analyze the multiple benefits of this approach. Then, departing from a single-objective early classification method proposed in [27], we propose a framework which deals with the problem of early classification of time series using multi-objective optimization techniques. For the first time, an early classification method will provide a set of solutions which will offer different trade-offs between accuracy and earliness in only one execution of the algorithm. This will enable the user to choose the solution which best suits his/her needs, in advance.

The rest of the paper is organized as follows. In Section 2, we define the problem of early classification of time series and introduce some other basic concepts and definitions. In Section 3, we present our proposal based on a multi-objective definition of the early classification problem. In Section 4 we perform extensive experiments to prove the validity of our approach and in Section 5 we summarize the obtained results. Next, in Section 6, we propose a real case study, based on the analysis of financial data from the Spanish Market, which will serve as an illustration of the applicability of the method. Finally, in Section 7 we draw the final conclusions and propose some research lines for the future.

2. Early classification of time series: problem setting

In order to properly define the problem of early time series classification, we must first define some basic concepts:

Definition 1. A *time series* is an ordered sequence of pairs (timestamp, value) of finite length L [36]:

$$TS = \{(t_i, x_i),\ \ i = 1, ..., L\}, \tag{1}$$

where the timestamps $\{t_i\}_{i=1}^{L}$ take positive and ascending real values and the values of the time series (x_i) take univariate or multivariate real values.

Definition 2. We will denominate $TS|_t$ to the time series TS, truncated, so that only the first t pairs are available, that is, $TS|_t = \{(t_i, x_i),\ \ i = 1, ..., t\}$.

Definition 3. Suppose we have a training set $X = \{(TS_1, CL_1), (TS_2, CL_2), \ldots, (TS_n, CL_n)\}$ of labeled time series, where TS_i are time series, and $CL_i \in \{1, 2, \ldots, k\}$ their respective class labels. *Time series classification* is a supervised learning task in which the objective is to build a mapping from the time series to their class labels by using X [2], which will be able to predict the classes of new unlabeled time series as accurately as possible (see Figure 1).

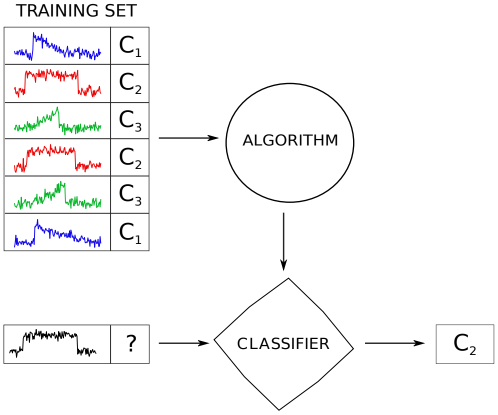


Figure 1: Supervised classification of time series.

With these definitions at hand, we are now able to understand the definition of early classification of time series as defined initially by [36]:

Definition 4. Suppose we have a training set $X = \{(TS_1, CL_1), (TS_2, CL_2), \ldots, (TS_n, CL_n)\}$ of labeled time series, where TS_i are time series, and $CL_i \in \{1, 2, \ldots, k\}$ their respective class labels. *Early time series classification* is a supervised learning task which attempts to build a mapping from the time series to their class labels by using X, which will be able to predict the classes of new unlabeled time series as early as possible, using only a part of the series $TS|_{t^*}$, but maintaining the accuracy that would be obtained if the whole time series were available.

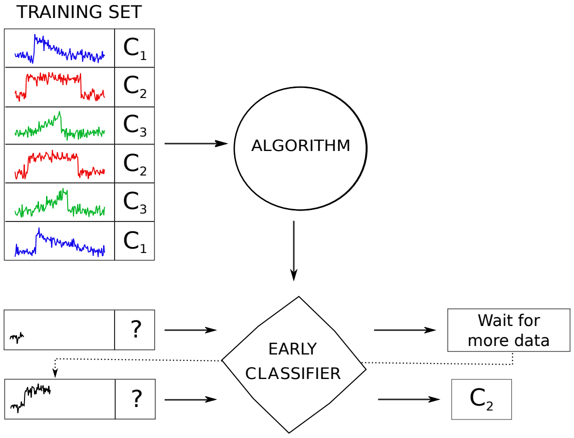


Figure 2: Early classification of time series.

As can be seen in Figure 2, usual early classifiers, do not always output a prediction. On the contrary, as new data becomes available, they evaluate if a reliable prediction can be made at that time, and if not, they abstain.

If we analyze the definitions and Figures 1 and 2, we can understand the difference between time series classification and early time series classification. In the first problem, the sole objective is to make accurate predictions, whereas in the second problem, we introduce the concept of earliness. It is quite logical to think that accuracy and earliness are usually conflicting objectives: the more data points are available from the time series, the more information we have and it is, thus, easier to make accurate predictions. On the contrary, if we want to make very early predictions, we will not have a lot of information about the series and so, it will be difficult to make accurate predictions.

In this context, early classification is of special interest when collecting additional measurements from the time series incurs in costs or when making late predictions has negative consequences [27]. Some examples are monitoring hospital patients and trying to identify crises as soon as possible [10], classifying different types of faults in an industrial plant [4], predicting stock crisis as early as possible [14] or trying to identify different bird species as early as possible using their songs, with the aim of automatically triggering recording devices [26].

In the original definition of the problem of early classification of time series proposed in [36], the accuracy which must be obtained is fixed in advance, and so the objective of many methods has focused essentially on minimizing the earliness. Typically, researchers have adopted this definition of the problem and, thus, they have given more importance to accuracy. However, this requirement is quite strict, and it is possible that, on many application domains, users are willing to sacrifice a bit on accuracy in order to obtain earlier predictions. Some recent solutions [6, 26, 27, 28, 32] include parameters or mechanisms to tune the weight associated to the accuracy and the earliness, finding different trade-offs between these two objectives. However, all these methods combine both objectives in some manner, formulating the problem as a single-optimization problem. The weight assigned to each objective must be defined in advance and, in order to obtain different trade-offs, the algorithms must be executed more than once with varying weight parameters. In summary, early classification of time series has never been dealt with using multi-objective methods, where all the possible trade-offs are sought simultaneously.

In this context, we propose a novel and more general definition of the problem of early classification:

Definition 5. Suppose we have a training set $X = \{(TS_1, CL_1), (TS_2, CL_2), \ldots, (TS_n, CL_n)\}$ of labeled time series, where TS_i are time series, and $CL_i \in \{1, 2, \ldots, k\}$ their respective class labels. *Early time series classification* is a supervised learning task in which the objective is to build a mapping from the time series to their class labels by using X, which will be able to predict the classes of new unlabeled time series as early and as accurately as possible.

This definition inevitably leads us to treat the problem as a multi-objective problem, where the goal is to optimize the costs of accuracy and earliness at the same time.

3. Early time series classification using multi-objective optimization techniques

In this section, we will introduce our early classification framework, which will be an extension of a previous approach presented in [27], which we will call *the baseline* method.

The aim of *the baseline* method was to obtain a pair $((h_1, h_2, ..., h_L), s_{\gamma *})$, where $(h_1, h_2, ..., h_L)$ is an ordered sequence of classifiers, one for each timestamp, and $s_{\gamma *}$ is a trigger function, optimized taking accuracy and earliness into account. The purpose of the classifiers is to output class predictions at each timestamp, and the trigger function will be in charge of deciding whether these class predictions should be considered or not.

Contrary to *the baseline method*, in our proposal, the goal is to obtain a set of pairs $\{((h_1, h_2, ..., h_L), s_{\boldsymbol{\gamma_1}}), ((h_1, h_2, ..., h_L), s_{\boldsymbol{\gamma_2}}), ..., ((h_1, h_2, ..., h_L), s_{\boldsymbol{\gamma_g}})\}$. In this case, instead of providing only one optimized trigger function, the method will yield a set of trigger functions, $\{s_{\boldsymbol{\gamma_1}}, s_{\boldsymbol{\gamma_2}}, ..., s_{\boldsymbol{\gamma_g}}\}$, all of them optimized considering accuracy and earliness, but based on the multi-objective perspective, explained in the previous section. Each of these trigger functions will acquire a different trade-off between earliness and accuracy. The user will then choose, based on the obtained results, one of these trigger functions based on the requirements of the problem at hand.

Both the *baseline method* and our proposal are built following three steps. First, a set of probabilistic classifiers are trained (Section 3.1). Then, in the second step, a specific shape for the trigger functions is chosen (Section 3.2). Finally, in the third step, shown in Section 3.3, these trigger functions are optimized. The first two steps will be identical in *the baseline method* and in our proposal; however, they will be explained in detail because they are essential aspects of both frameworks. The main novelty of our approach lies in the third step: the *baseline method* will formulate and solve a single-objective optimization problem while in this work an alternative multi-objective formulation is introduced. Both approaches will be presented for comparison and the benefits of using our proposal will also be discussed in this section.

3.1. Training the classifiers

The first step is to obtain a set of classifiers which will be used to obtain class-predictions at each timestamp. Recall that this step is identical in *the baseline method* and our proposal. As formulated above, we will train a set of classifiers $(h_1, h_2, ..., h_L)$, one for each timestamp, using the training set of labeled time series X, following the procedure shown in Figure 3.

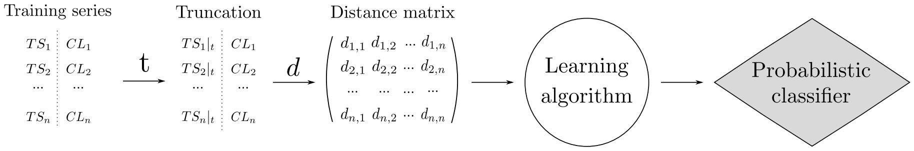


Figure 3: Construction of the probabilistic classifiers

As can be seen, in order to build a classifier for timestamp t, we take all the training series in X and truncate them to this timestamp. Note that this timestamp t can be defined as an absolute value, or as a percentage of the length of the series. This second formulation must be used, for example, if the database contains series of different lengths. Then, instead of building the classifier using the raw time series, we build a distance matrix first by using a distance measure d of choice.

This technique has been used in studies such as [19, 27, 26] and it allows to include specific time series distance measures such as Dynamic Time Warping (DTW) [29] into the framework. Since typical learning algorithms for classification require an input of a fixed dimension, learning with the distance matrix, instead of the raw values of the series, allows working with databases which contain series of different lengths. For this, we must simply choose a distance measure which is able to compare series of different lengths, such as DTW. Additionally, learning with the distance matrix also has many other benefits such as including time-flexibility into the classification framework, dealing with noise or outliers, etc. [1]. Finally, the classifier is trained using this distance matrix as input.

In order to train the classifiers, we can choose any learning algorithm of our choice. The only requirement, which will be understood better in the following sections, is that the classifier must be able to output class-probabilities (probabilities of membership to each class), instead of outputting only a class prediction.

3.2. Definition of the trigger function

The next step is to define the trigger functions. As mentioned earlier, and as can be seen in Figure 4, the trigger functions will be responsible for deciding whether a class prediction issued by a given classifier h_t at a given timestamp t is reliable and should be considered, or we should not trust it, and wait for more data. For making this decision, the trigger functions will use certain information regarding the prediction. In this case, as can be seen in Figure 4, we will use the class-probabilities, $\mathbf{p^t} = (p_1^t, p_2^t, ..., p_k^t)$ outputted by the selected classifier h_t, at the given timestamp t, together with the timestamp in which the prediction is made (t). The main reason for using the class-probabilities is that we suppose that the distribution of the class probabilities over time will provide us information about the reliability of the prediction. Indeed, this intuition has shown to be well-founded in previous studies [26, 27], so it has been directly adopted.

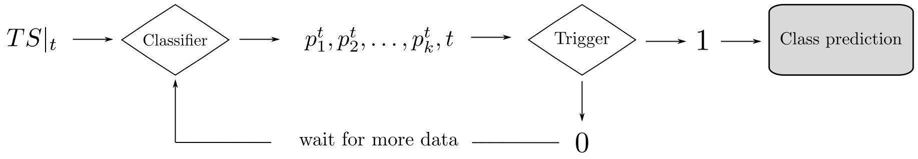


Figure 4: Schema of early classification framework

As can be seen in Figure 4, the trigger function outputs a value of 0 or 1. If the trigger function decides that the prediction is reliable (output 1), then, the class prediction is made by choosing the class with the highest predicted probability. On the contrary, if the trigger function determines that the prediction is not yet reliable (output 0), then we will have to wait until we collect more data.

With these input variables we could construct many different trigger functions, with different shapes. However, we have decided to use a linear stopping rule of the following shape, which was proposed as part of the *the baseline method* in [27]:

$$s_{\gamma}(\mathbf{p^t}, t) = \begin{cases} 0 & if \quad \gamma_1 p^t_{1:k} + \gamma_2 (p^t_{1:k} - p^t_{2:k}) + \gamma_3 \frac{t}{L} \leq 0 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where $p^t_{1:k}$ and $p^t_{2:k}$ are the first and second largest probabilities obtained at time t and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)$ is a vector of parameters with $\gamma_i \in [-1, 1]$, which has to be chosen.

This trigger function has shown a competitive performance when used within *the baseline method* in comparison to other more complex rules and has outperformed the results of the state-of-the-art methods [27]. As such, as in the *baseline method*, all the trigger functions obtained by our proposal will have this shape, and the difference between them will be the $\boldsymbol{\gamma}$ selected in each case.

Now, a method must be designed to select the $\boldsymbol{\gamma}$ parameters which will allow us to obtain optimal results in terms of accuracy and earliness.

3.3. Finding the optimal set of trigger functions

As mentioned, in order to completely define a trigger function defined as in Equation 2, the $\boldsymbol{\gamma}$ parameter vector must be chosen. If we substitute the γ_i parameters by randomly chosen values in $[-1, 1]$, the trigger function may already be used. However, probably, the classification results that will be obtained with this trigger function will not be optimal in terms of earliness and accuracy. But what does it mean to be optimal in terms of earliness and accuracy? First we must find a manner to evaluate a given trigger function in terms of these two objectives, and then, we can focus on finding the optimal one or ones.

3.3.1. Evaluation of a trigger function

To evaluate a trigger function in terms of accuracy and earliness, we will use two evaluation measures, which have been previously used in all the previous early classification studies for evaluation purposes. On the one hand, the evaluation of the earliness, will be carried out using the following measure:

$$C_e(X, s_\gamma) = \frac{1}{|X|} \sum_{x \in X} \frac{t_x^*}{L_x} \cdot 100 \tag{3}$$

t_x^* being the earliest timestamp in which a class-prediction is obtained for series x, and L_x being the length of x. In our case, t_x^* corresponds to the first time that the chosen trigger function s_γ outputs a value of 1. As can be seen, this measure measures the average time that we must wait to obtain class-predictions for the series in X with trigger function s_γ, and it is represented as a percentage of the length of the series in X.

On the other hand, the accuracy is typically evaluated by calculating the classification error, or percentage of incorrectly classified time series:

$$C_a(X, s_\gamma) = \frac{1}{|X|} \sum_{x \in X} \mathbb{I}(\hat{CL_x} \neq CL_x) \cdot 100. \tag{4}$$

where $\hat{CL_x}$ is the predicted class at instant t_x^* for time series x, and CL_x is its true class value. $\mathbb{I}(\cdot)$ takes a value of 1 if the condition is true, and 0 otherwise. In our case, since the classifiers output class-probabilities, $\hat{CL_x}$ corresponds to the class with the highest assigned probability at instant t_x^*.

In order to calculate $C_e(X, s_\gamma)$ and $C_a(X, s_\gamma)$ for a given trigger function we will use the training set of time series X. Note that, in order to calculate C_e, we must obtain the first timestamp in which the trigger function s_γ would return 1 for each of the series in X. For this, we need the class-probabilities of all the series in X at all the possible timestamps.

To obtain these class-probabilities we could use the $(h_1, h_2, ..., h_L)$ classifiers presented in Section 3.1, but recall that these were trained by using the same training set X. So if we use these classifiers to evaluate the trigger functions, we might overfit and the results might be overly optimistic. Instead, we have followed a 5-fold cross-validation procedure: the training set is divided into 5 folds, and 5 classifiers are built at each timestamp by leaving one fold out in each iteration. The procedure used to build these classifiers will be identical to that followed to train $(h_1, h_2, ..., h_L)$. Then, the class-probabilities for each $x \in X$ are obtained with a classifier which has not seen this instance before, and, thus, we avoid overfitting.

3.3.2. Optimization of the trigger functions

Now that we know how to evaluate a specific trigger function in terms of accuracy and earliness, we can focus on trying to find a trigger function which minimizes these two objectives simultaneously. However, as mentioned on more than one occasion, earliness and accuracy are conflicting objectives. So, how do we optimize them simultaneously?

In *the baseline method* [27], the authors proposed a single-objective formulation of the problem which combined the measures C_e and C_a in a unique linear cost function, where each of the objectives was weighted by a user-defined parameter α which ranged between 0 and 1. Once this cost function was defined, it was minimized using a genetic algorithm.

With this approximation, *the baseline method* only provides one solution, subject to a certain parameter (α) that the user must define in advance. Something similar happens in every other early classification method proposed in the literature until now. However, which early classifier is better, one which obtains an accuracy of 80% and predicts using only 20% of the length of the series, or a solution which yields an accuracy of 90% but needs 30% of the length of the series? It depends on the needs and requirements of the application area at hand and the interests of the user, but, other than that, we can not say that one is better than the other. Indeed, even if the user has a clear idea of the accuracy and earliness values that the problem requires, this α parameter does not provide any intuition regarding the exact accuracy or earliness values that will be obtained by the method, only the importance that will be given to each objective. In this sense, it is quite difficult to choose a specific value in advance.

The Pareto Optimality criterion states that one solution is better than another (dominates), if it is better in at least one objective, while obtaining at least equal solutions in all the rest of the objectives. Thus, the goal is to find the set of non-dominated solutions also called Pareto optimal solutions, which dominate all the rest of the solution space.

So, if the early classification problem can be seen as a multi-objective optimization problem, why not try to obtain all the possible non-dominated trigger functions at the same time? Then, the user can choose the most suitable on each occasion based on the obtained solutions, instead of having to decide beforehand. Precisely in this aspect lies the main novelty of the proposed method: for the first time, we will solve the problem of early classification using techniques from the multi-objective optimization area of knowledge. As such, instead of finding one optimal trigger function s_{γ^*}, we will find a set $\{s_{\gamma_1}, s_{\gamma_2}, ..., s_{\gamma_g}\}$, which will provide non-dominated results in accuracy and earliness. For this, we will consider the cost functions defined in Equations 3 and 4, simultaneously and separately and we will try to solve the following optimization problem:

$$\min_{\gamma}(C_e(X, s_{\gamma}), C_a(X, s_{\gamma})) \tag{5}$$

As in *the baseline method* [27], in this case, we will also solve this multi-objective optimization problem using meta-heuristic algorithms due to the complexity of the two objective functions.

The advantages of the multi-objective approach are evident. Firstly, if we combine the two cost function as in *the baseline method*, C_a and C_e must be scaled into the same interval, so that one of them does not dominate the search. Secondly, in the single-objective approach, the weight assigned to each of the objectives has to be decided in advance, and the α parameter is usually not easy to choose, since the two cost functions do not vary in the same manner and its meaning is different in each database. Finally, in order to obtain various solutions with different trade-offs between the two objectives, in the single-objective case, the algorithm must be executed more than once, whereas in the multi-objective case a set of non-dominated solutions is obtained in only one execution. The user can decide with much more information at hand, which solution is the most suitable for its needs. All these arguments could be used similarly with other state-of-the art methods such as [26, 28] which also include parameters which allow to tune the trade-off between the two objectives.

3.4. Usage of the proposed early classification framework

Now that we know how our early classification framework can be built, we will remember what type of solutions are obtained and how they can be used. As mentioned, in our proposal, we will obtain a set of pairs $\{((h_1, h_2, ..., h_L), s_{\gamma_1}), ((h_1, h_2, ..., h_L), s_{\gamma_2}), ..., ((h_1, h_2, ..., h_L), s_{\gamma_g})\}$. So, instead of having only one trigger function the method will output a Pareto set of trigger functions. For example, in Figure 5, each of the points represents one of the trigger functions in the Pareto set, obtained by the proposed multi-objective early classification framework. The accuracy and earliness values obtained by this trigger function are represented by its values in the two coordinate-axes. For example, the solution shown in red is a trigger function which obtains a 65$\%$ of accuracy approximately and uses the 13$\%$ of the length of the time series to make predictions on average. Note, of course, that these performance values are calculated on the training set, which is the dataset we use to build the Pareto set. As can be seen, the methodology, provides us with a set of solutions with different trade-offs between the two objectives.

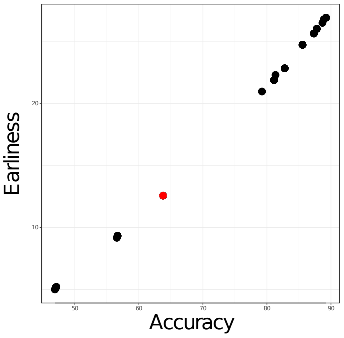


Figure 5: Example of Pareto front obtained by our proposal for an early classification problem.

Once the Pareto set of solutions is obtained, the user can choose the trigger function (s_{γ^*}) which best fits to its accuracy and earliness requirements by analyzing the different solutions (see Figure 6). Of course, we must emphasize again that the results shown to the user are calculated on the training set, and, thus, it can not be guaranteed that the same exact values will be obtained for new time series. However, if the training and testing sets are independent but similarly distributed, the approximation should be good enough to make this decision and they can be used as a reference when choosing one solution over another. In this context, given the Pareto set in Figure 5, a user with high accuracy requirements will choose a solution which is located on the right top-side of the graphic and, on the contrary, a user with high earliness requirements will choose a solution on the left bottom-side of the graphic.

Once a specific solution is chosen, we can use this trigger function together with the $(h_1, h_2, ..., h_L)$ classifiers to make early class predictions of new time series using the procedure shown in the gray rectangle of Figure 6.

4. Experiments

In this section, we present the experimental setup used to evaluate our multi-objective early classification framework, we define the parameter settings of our method and the state-of-the-art methods we will use for comparison, and we explain how the early classification proposals will be evaluated.

4.1. Data

The UCR archive [5], is a large repository of time series databases, specifically prepared for experimenting with new time series classification and clustering methods. Most of the previous early classification methods, as well as most

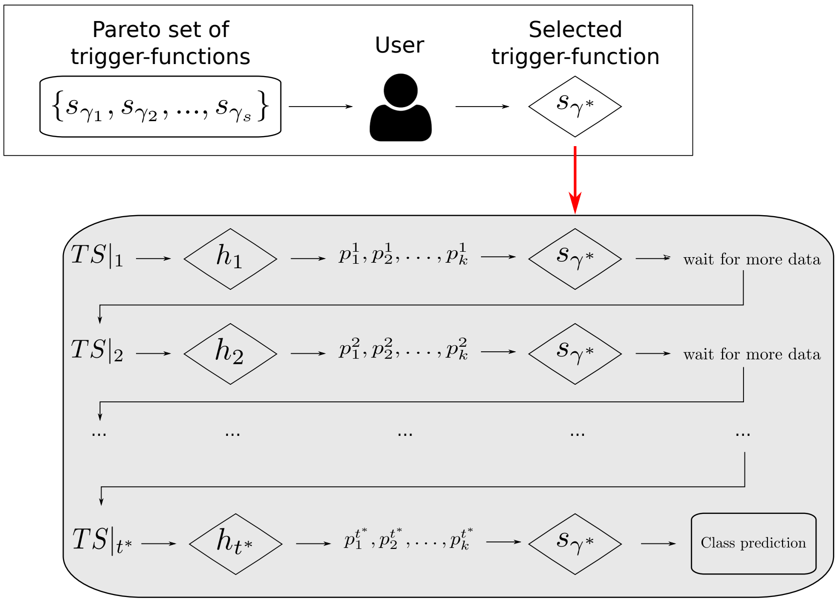

Pareto set of trigger-functions
$\{s_{\gamma_1}, s_{\gamma_2}, ..., s_{\gamma_s}\}$
User
Selected trigger-function
s_{γ^*}
$TS|_1$
h_1
$p_1^1, p_2^1, \ldots, p_k^1$
s_{γ^*}
wait for more data
$TS|_2$
h_2
$p_1^2, p_2^2, \ldots, p_k^2$
s_{γ^*}
wait for more data
... ...
$TS|_{t^*}$
h_{t^*}
$p_1^{t^*}, p_2^{t^*}, \ldots, p_k^{t^*}$
s_{γ^*}
Class prediction

Figure 6: Usage of the multi-objective early classification framework

time series classification and clustering proposals are evaluated and tested on these datasets.

In this experimentation 45 databases from this archive have been used, which will be specified in the following sections. These databases have been chosen because results for various early classification state-of-the-art methods, which we will use for comparison, are available and published for these 45 databases. All these datasets are conformed of univariate time series and the lengths of the series within each database are equal. However, the datasets are obtained from different contexts, they are of different dimensions and they contain variate number of classes. Additionally, bear in mind that the framework can also be used for databases which contain series of different lengths. For this, the step-size for learning the classifiers must be defined as a percentage of the length of the series, and a suitable distance measure such as DTW must be used (see Section 3.1).

4.2. Comparison with other state-of-the-art methods

To begin with, we have a special interest in comparing our methodology with the performance of the method from which we have departed: *the baseline method* [27]. The main reason for this is that the two methods are both based on cost minimization techniques, but the baseline method proposes a solution based on single-objective optimization, while this work proposes a multi-objective generalization. The idea is to directly analyze if the multi-objective approach provides better and more useful results than the single-objective solution approach.

The parameter settings of *the baseline method* are set based on the original paper. Firstly, the posterior probabilities are obtained from Gaussian Process classifiers, using an extension of the *vbmp* package in R [21], since these classifiers report the best results in the study. The classifiers are built every $\%5$ of the length of the series, as in the original paper, choice which will be discussed more in depth in the following section. The optimization is carried out using the genetic algorithm with default parameters implemented in the *GA* package of R [30]. A real-valued vector codification is used and the initial population is chosen uniformly at random. Selection is carried out by using fitness proportional selection with linear scaling, arithmetic crossover operator is applied with a probability of 0.8, and mutation is performed with a probability of 0.1 using the nonuniform random mutation operator. At each iteration $(100$ in total), 5% of the best individuals from the initial population will survive and will replace the worst 5% obtained from the genetic operations. Next, the α parameter, which tunes the trade-off between earliness and accuracy has to be chosen. In this case, since the objective is to compare these solutions with the set of solutions obtained by the multi-objective framework, we will select a more comprehensive set of α values, in comparison to the original work, which only considered values which benefited the accuracy. Indeed, we will consider $\alpha \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75,$ $0.8, 0.85, 0.9, 0.95\}$. Finally, since the genetic algorithm has a random component, for each α, we have executed the algorithm 30 times, as in the original experimentation, and in the last step, we have chosen the solution with the median cost.

Additionally, to further validate our proposal, we have also compared it with the four most relevant state-of-the-art early classification methods with available source codes: ECTS [1], EDSC [2] Rel.Class.[3] and ECDIRE [4]. These methods have different variants and the selection of the parameters for these methods has been made based on the original papers (see Table 1 for details). Note that for some methods we obtain only one solution per database, and for others, due to different

[1] http://zhengzhengxing.blogspot.com.es/p/research.html

[2] http://zhengzhengxing.blogspot.com.es/p/research.html

[3] http://www.mayagupta.org/publications/Early_Classification_For_Web.zip

[4] http://www.sc.ehu.es/ccwbayes/members/umori/ECDIRE/ECDIRE.html

parameter configurations, we obtain more than one. However, in all cases, the methods give more importance to accuracy, so the obtained solutions tend towards high accuracy values.

Method	Variants	Parameter name	Values
ECTS [36]	-Strict -Loose	Minimum support	0, 0.05, 0.1, 0.2, 0.4, 0.8
EDSC [37]	Chebyshev Inequality version	Chebyshev bound	2.5, 3, 3.5
RelClass [28]	-Naive Gaussian Quadratic set -Gaussian Naive Bayes box	Reliability threshold τ	0.001, 0.1, 0.5, 0.9
ECDIRE [26]		acc_perc	100%

Table 1: Combinations and variants of the comparison methods.

4.3. Parameter settings of our proposal

In order to completely define our early classification framework, we must make two important decisions. On the one hand, we must select a multi-objective optimization algorithm to solve the problem. On the other hand, we need to select a learning algorithm to build the classifiers (see Figure 3).

With regards to the first aspect, we have mentioned that we will use meta-heuristic algorithms. Analyzing the performance of different evolutionary algorithms to solve this task is beyond the scope of this paper, and, indeed, different algorithms will probably obtain varying solutions in different databases. In this context, in order to perform our experimentation, we have chosen the NSGA II optimization algorithm [7], a fast and elitist multi-objective genetic algorithm which is one of the most popular multi-objective meta-heuristics. Note that we have made this choice simply based on the popularity of the algorithm and in order to establish a parallelism with *the baseline method*, which also uses genetic algorithms. However, this is a parameter of the framework and, thus, the choice of the evolutionary algorithm could be modified by the user when looking for the best performing configuration for a particular problem or database.

The NSGA II algorithm, has been implemented using the *mco* package from R [25]. As in the baseline method, the parameter settings of the optimization method have not been tuned, we have used the default values of function *nsga2* of R. As in the baseline method, a vector representation is used and the initial population is chosen randomly. Then, the default genetic operators and parameters of the *nsga2* function are applied: binary tournament selection, binary simulated crossover and polinomial mutation with parameters $cprob = 0.7$ (crossover probability), $mprob =$

0.2 (mutation probability), $cdist$ = 5 (crossover distribution index) and $mdist$ = 10 (mutation distribution index). Only the population size has been changed to 56 for similarity with *the baseline method*, which uses a population size of 50 by default [5]. Note that the population size will influence the final number of trigger functions provided by the method. Since genetic algorithms are randomized heuristics, we have executed the NSGA II algorithm 30 times, as in *the baseline method*. Since in the multi-objective case it does not make sense to calculate the median cost of the obtained solution set, we have chosen the solution set with the median hypervolume, which is an evaluation measure for multi-objective solutions, which will be explained more in detail in the next section. As with the choice of the evolutionary algorithm, the choice of the parameters of the evolutionary algorithm could also be adapted to each database or fine-tuned, also possibly obtaining even better results.

In relation to the second decision, the choice of the classifiers, we have selected the same type of probabilistic classifiers used in the *baseline method*, Gaussian Process classifiers trained using *vbmp*, also with the objective of being fair in the comparisons.

Also following the choices made in *the baseline method*, the classifiers are built every 5% of the length of the series, instead of on each timestamp. This is mainly done in order to limit the computational burden associated to the experimentation. Additionally, the reason for choosing a percentage of the length as the step-size and not an absolute number of data-points is that the experimentation has been carried out using many different databases, each of them containing series of very different lengths (some contain series of only 24 data-points whilst others contain series of more than 1000 data-points). In this context, choosing an absolute value which is valid for all databases and which is affordable in terms of computational time is not possible. Nevertheless, when applying this framework to only one database, this step-size could be better tuned, choosing an absolute number of data points, or even considering uneven sampling rates, based on domain knowledge.

Finally, since the databases that we use in the experimentation do not contain series of different lengths, the distance matrices, which will be the input to the classifiers are built using the Euclidean distance. This distance measure was also used in the *baseline method* and we also maintain it to be fair in the comparative.

[5] The *nsga2* function of package *mco* requires the population size to be a multiple of 4, for here the small difference with *the baseline method*.

However, remember that if our database contained series of different lengths, the framework could be easily accommodated by simply choosing another distance measure which is capable of working with this type of database, such as DTW.

4.4. Evaluation

As in previous early classification studies, we have used the train/test evaluation framework, using the pre-defined splits provided in the UCR archive. This means that the early classification framework is built using training set X and then, these solutions are applied to an independent testing set.

In order to measure the quality of the solutions, we use the two costs associated to the problem: C_e and C_a defined in equations 3 and 4. However, since we want to take into account the bi-objective nature of the problem, we will use different evaluation measures obtained from the multi-objective optimization area of knowledge:

- **Number of non-dominated solutions:** these values will be obtained by following the Pareto optimality criterion, which as mentioned previously tells us that a solution dominated another if it improves in at least one objective, while obtaining equal results in the others. We will compare the methods pairwise: we will consider all the solutions obtained by both methods, we will evaluate them on the testing set, and finally, we will count the number of non-dominated solutions obtained by each of the methods. See Figure 7, to view an example of this process: in the first figure, the solutions obtained by the baseline method for a specific database and evaluated on the testing set are plotted; in the second figure, the same process is followed for our proposed method; in the last figure, we combine the results of the two methods and plot only the non-dominated ones.

  Since the number of solutions provided by each method is different, we will base our analysis regarding domination counts on the proportion of non-dominated solutions.

- **Spread and diversity:** In addition to obtaining good results in terms of domination counts, we are also interested in obtaining well-spread solution sets, which will take over a large part of the search space and which are as diverse as possible. To measure the goodness of the obtained solution sets regarding these aspects we use different measures. To begin with, we calculate the Δ method as proposed in [7]:

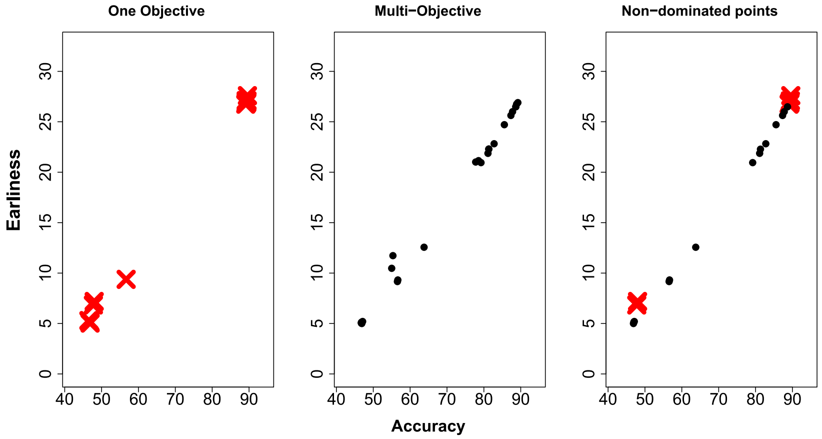


Figure 7: Computation of domination counts for the CBF database from the UCR, when comparing the baseline method with our proposal

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \tag{6}$$

where d_l and d_f are the Euclidean distances to the two extreme solutions, the d_i values are the Euclidean distances between neighboring solutions, and $\bar{d}$ is the mean of all the d_i values (see Figure 8). This measure takes a value of 0 in the best possible case, when all the solutions are distributed uniformly over the solution space. This measure will only be considered when the number of solutions is higher than 2.

Additionally, we calculate the maximum spread, M_3 as the Euclidean distance between the two most extreme solutions (see solutions shown in red in Figure 8). Logically. we will only calculate this measure when the number of solutions is at least 2.

- **Dominated hypervolume**: this measure measures the hypervolume of the space that a solution set dominates with respect to a reference point [7] (see grayed area in Figure 8). A solution set which yields a higher hypervolume is considered better, since it dominates a larger part of the solution space and covers a larger part of it. In this sense, the hypervolume is a quite general measure, which measures both the quality and the diversity of the solution set. In our case, we have used the hypervolume calculator from the *mco* package [25]. As can be seen in Figure 8 we have taken (100,100) as the reference point, since this is the worst possible solution, considering that we are minimizing costs C_a and C_e. Based on its formulation, we will only calculate this measure when the number of solutions is at least 2.

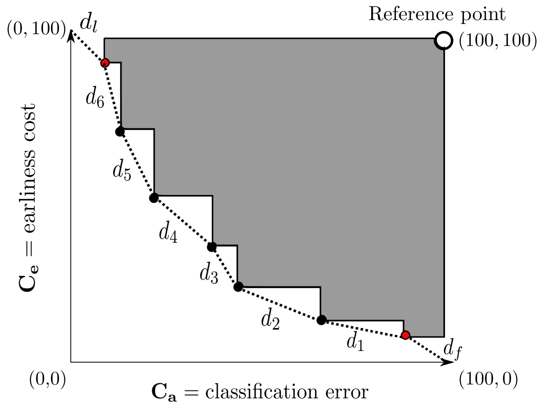


Figure 8: Performance measures for multi-objective optimization.

Note that the hypervolume and the spread will only be calculated when comparing to the baseline method. The reason is that the rest of the methods are aimed at obtaining high accuracy values, and so, they obtain very few solutions in most cases, all situated in a specific zone of the solution space. Additionally, as can be seen in Figure 7, we must emphasize that the solutions, when applied to the testing set, might not all be non-dominated. However, when calculating the spread metrics and the hypervolume, we will remove the dominated solutions within each method.

5. Results

To begin with, in Table 2 we show the proportions of non-dominated solutions obtained when comparing our method and the other state-of-the-art methods pairwise. These are calculated following the procedure shown in Figure 7. In each case, we show the number of non-dominated solutions/the total number of solutions that the corresponding the method provides. The first column always refers to our method and the second to the competitor which is named in the column header. In bold we highlight the highest proportion in each comparison. Additionally, we have carried out a paired Wilcoxon test with an alternative hypothesis stating that the mean proportion of non-dominated solutions obtained by our

Table 2: Number of non-dominated solutions obtained from pairwise comparisons between methods. For each of the competing methods, the first column refers to our multi-objective method and the second column to the competitor. Each pair of numbers represents the following: number of non-dominated solutions of the method/ number of total solutions of the method. In bold we show the method which obtains a higher proportion of non-dominated points for each pairwise comparison.

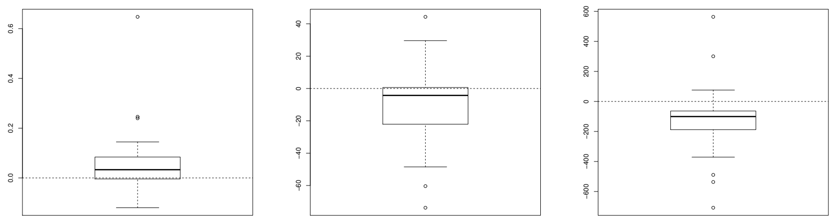
	Baseline		**ECDIRE**		**Rel.Class.**		**ECTS**		**EDSC**	
50words	**32/51**	9/18	**34/51**	0/1	**34/51**	0/8	**34/51**	0/12	**34/51**	0/3
Adiac	**24/49**	5/17	28/49	**1/1**	**28/49**	4/8	**28/49**	0/12	**28/49**	0/3
Beef	3/12	**4/10**	**5/12**	0/1	1/12	**5/8**	5/12	0/12	**5/12**	0/3
CBF	14/26	**7/17**	16/26	**1/1**	**16/26**	3/8	**16/26**	0/12	**16/26**	1/3
ChlorineConcentration	**20/37**	0/12	**20/37**	0/1	**20/37**	3/8	**20/37**	0/12	**20/37**	0/3
CinC_ECG_torso	**19/25**	6/18	**20/25**	0/1	**20/25**	3/8	0/25	**6/12**	**20/25**	0/3
Coffee	**5/13**	1/5	6/13	**1/1**	2/13	**4/8**	**6/13**	0/12	**6/13**	0/3
Cricket_X	30/50	**11/17**	**34/50**	0/1	**34/50**	1/8	**34/50**	0/12	**34/50**	0/3
Cricket_Y	**35/50**	8/16	**37/50**	0/1	**37/50**	0/8	**37/50**	0/12	**37/50**	0/3
Cricket_Z	**36/52**	9/18	**40/52**	0/1	**40/52**	2/8	**40/52**	0/12	**40/52**	0/3
DiatomSizeReduction	5/12	**1/1**	5/12	**1/1**	**5/12**	2/8	**5/12**	2/12	**5/12**	1/3
ECG200	**6/13**	3/8	**7/13**	0/1	**7/13**	0/8	**7/13**	0/12	**7/13**	0/3
ECGFiveDays	**9/14**	2/11	**9/14**	0/1	**9/14**	2/8	**9/14**	3/12	9/14	**2/3**
FaceAll	**37/48**	12/18	**39/48**	0/1	**39/48**	0/8	**39/48**	0/12	**39/48**	0/3
FaceFour	**7/12**	3/9	**7/12**	0/1	**7/12**	0/8	**7/12**	0/12	**7/12**	0/3
FacesUCR	**34/50**	11/17	**37/50**	0/1	**37/50**	0/8	**37/50**	0/12	**37/50**	0/3
fish	**24/42**	8/15	**25/42**	0/1	**25/42**	0/8	**25/42**	0/12	**25/42**	0/3
Gun_Point	**11/29**	2/12	**11/29**	0/1	**11/29**	0/8	**11/29**	0/12	11/29	**2/3**
Haptics	**11/32**	4/15	**14/32**	0/1	**14/32**	2/8	**14/32**	0/12	**14/32**	0/3
InlineSkate	5/24	**4/16**	**5/24**	0/1	**5/24**	0/8	5/24	**4/12**	**5/2**4	0/3
ItalyPowerDemand	**12/26**	8/18	15/26	**1/1**	**15/26**	3/8	**15/26**	3/12	**15/26**	0/3
Lighting2	**2/6**	2/7	**3/6**	0/1	**3/6**	0/8	**3/6**	0/12	**3/6**	1/3
Lighting7	**7/16**	3/10	**7/16**	0/1	**7/16**	3/8	**7/16**	2/12	**7/16**	1/3
MALLAT	**34/40**	11/18	**35/40**	0/1	**35/40**	1/8	**35/40**	1/12	**35/40**	0/3
MedicalImages	**17/37**	4/16	**19/37**	0/1	**19/37**	0/8	**19/37**	0/12	**19/37**	0/3
MoteStrain	**19/32**	6/16	**19/32**	0/1	**19/32**	2/8	**19/32**	2/12	**19/32**	0/3
OliveOil	**3/10**	1/7	**3/10**	0/1	1/10	**4/8**	**3/10**	1/12	3/10	**1/3**
OSULeaf	**11/35**	3/14	**12/35**	0/1	**12/35**	0/8	**12/35**	0/12	12/35	**2/3**
SonyAIBORobotSurface	**1/3**	1/7	1/3	**1/1**	**1/3**	1/8	**1/3**	0/12	**1/3**	0/3
SonyAIBORobotSurfaceII	**6/13**	6/17	**7/13**	0/1	**7/13**	2/8	**7/13**	3/12	**7/13**	1/3
StarLightCurves	**36/51**	6/17	**39/51**	0/1	**39/51**	0/8	**39/51**	0/12	**39/51**	0/3
SwedishLeaf	**28/48**	7/15	30/48	**1/1**	**30/48**	0/8	**30/48**	0/12	**30/48**	0/3
Symbols	**4/13**	3/17	5/13	**1/1**	**5/13**	2/8	**5/13**	3/12	**5/13**	0/3
synthetic_control	**27/45**	8/16	25/45	**1/1**	**25/45**	3/8	**27/45**	0/12	**27/45**	0/3
Trace	3/14	**5/9**	**4/14**	0/1	3/14	**4/8**	**4/14**	0/12	3/14	**3/3**
TwoLeadECG	6/17	**8/18**	8/17	**1/1**	8/17	**4/8**	**8/17**	0/12	8/17	**2/3**
Two_Patterns	45/52	**16/18**	47/52	**1/1**	**41/52**	4/8	**44/52**	3/12	**40/52**	2/3
uWaveGestureLibrary_X	**49/56**	12/18	51/56	**1/1**	**51/56**	0/8	**51/56**	0/12	**51/56**	0/3
uWaveGestureLibrary_Y	**47/55**	15/18	49/55	**1/1**	**49/55**	0/8	**49/55**	0/12	**49/55**	0/3
uWaveGestureLibrary_Z	47/53	**17/17**	**48/53**	0/1	**48/53**	0/8	**48/53**	0/12	**48/53**	0/3
wafer	**13/23**	8/17	**15/23**	0/1	**15/23**	2/8	**15/23**	4/12	**15/23**	0/3
WordsSynonyms	**30/42**	8/18	**32/42**	0/1	**29/42**	2/8	**30/42**	7/12	**32/42**	0/3
yoga	22/37	**13/17**	27/37	**1/1**	**27/37**	0/8	**27/37**	0/12	**27/37**	0/3
NIFECG_Thorax1	32/46	**12/17**	**34/46**	0/1	**34/46**	0/8	**34/46**	0/12	**34/46**	0/3
NIFECG_Thorax2	**21/47**	8/18	**28/47**	0/1	**28/47**	0/8	**28/47**	0/12	**28/47**	0/3

method is larger than that obtained by each competitor, and we have obtained a p-value of $6.170133 \cdot 10^{-4}$ for the baseline method, and p-values of $1.117522 \cdot 10^{-3}$, $2.275016 \cdot 10^{-6}$, $5.577202 \cdot 10^{-8}$ and $1.271760 \cdot 10^{-6}$ for the rest of the methods respectively, all in favour of our method. In fact, we must emphasize, that these p-values are very low, and that the proportion of non-dominated solutions of the competitors, except maybe the baseline method, are very frequently 0, which proves that our methodology dominates all the solutions proposed by these methods, even if these are designed to obtain very high accuracy values. We have also performed plots, similar to Figure 7 for all the 45 datasets and all the considered competitors, where the exact non-dominated solutions can be seen. Due to the lack of space we do not include these figures in the paper, but they are made available in our website [6].

In addition to the domination counts, we provide a more detailed comparison with *the baseline method* to further validate the goodness of our approach. Recall that we are interested in obtaining uniform spread and diverse solutions. To analyze this aspect, in Figure 9 we represent the differences obtained in Δ, M_3 and hypervolume when comparing the baseline method and our proposed method in the 45 datasets (the subtraction is done in this order). As can be seen, for the first evaluation measure, Δ, the box lays just over the 0 value, so most differences are positive. Recall that this value measures the uniformity of the spread of the solutions over the solution space, and smaller values, represent better spread solutions. As such, we can say, that in most cases our method obtains more suitable solution sets in this sense. The second evaluation measure, M_3, measures the distance between the two more distant solutions, and so, higher values would indicate better solution sets. In this case, the box lays just below the 0 value, and so, in most cases, our proposal obtains higher M_3 values than the baseline method. Finally, in the case of the hypervolume, the entire box lays below the 0 value. This indicates that in most cases, our method obtains a higher hypervolume, and thus a better solution set. The actual values obtained for these measures by the baseline method and our proposal are available as supplemental material in our website. Additionally, we have also performed a Wilcoxon test on these differences, to test whether their mean is greater than 0 ($H_1 : \mu > 0$) in the case of the Δ parameter or lower than 0 ($H_1 : \mu < 0$) for the M_3 and hypervolume parameters, and we have obtained p-values of 0.001679, 0.002093 and $3.827 \cdot 10^{-7}$, respectively, in favor of our proposal. As such, the multi-objective approach outperforms the

[6] http://www.sc.ehu.es/ccwbayes/members/umori/ECMulti/ECMulti.html

single-objective approach also in diversity and spread.



(a) Δ (b) M_3 (c) Hypervolume

Figure 9: Boxplots of differences between evalution measures values obtained by the baseline method minus the evaluation measures values obtained by our proposed method.

6. Case study: Ibex 35

In order to show the usefulness of the proposed multi-objective approach in a real context, we have carried out a case-study in the field of economics.

In spite of the predominance of the theory of Efficient Markets [11], which states that the prices of the stocks have a random behaviour and that the precision of its prediction can not exceed 50% [3], there are various theoretical and practical indications that state that this leads to an inefficient financial market [24]. Indeed, the stock market has a long memory, even long term (Hu et al, 2015), which implies that the prices of financial assets have a predictable component that repeats certain patterns over time. Detecting them in advance would lead to a superior performance and subsequently to beat the stock market.

Specially in the past decades there have been abundant attempts to try to model the behavior of financial assets using not only convencional statistical methodologies but also machine learning [20, 34]. However, despite all the hard work, prediction of the stock index remains a defiant matter due to the amount of factors that interact in the market [35] and the complex, highly noisy, dynamic, nonlinear, nonparametric, high dimensional and chaotic nature of this type of data [8].

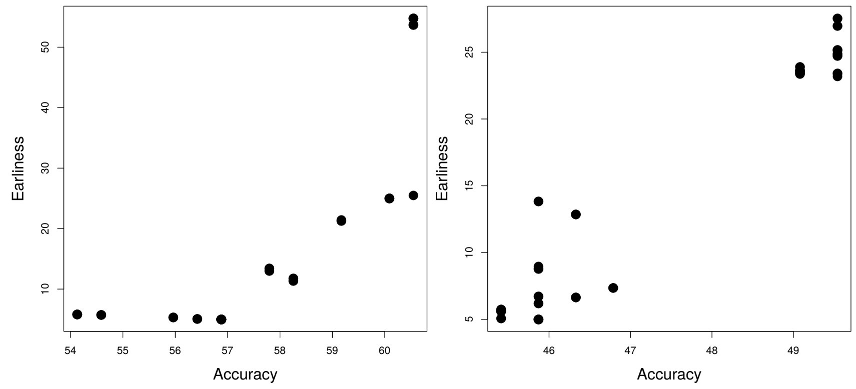
The goal of these experiments is to apply the early classification framework explained above to predict, as soon as possible, whereas a specific financial index will increase or decrease during a given trading session. Based on the obtained results and since stock prices direction prediction is a key reference for designing better trading strategies [18, 22], it would be possible to generate advanced indicators that could be used in the trading systems to give advice of when to buy and sell during a session and to create a system to alert investor of the markets sudden movements.

The data used in the experimentation is obtained from the Spanish Market, and the measurements correspond to the intra day trading prices of the index traded Ibex 35 obtained from BME Market Data, which belongs to the "Grupo de Bolsas y Mercados Españoles" (The Spanish Stock Exchange), specialized in the processing, generating and commercializing the information that originates for the different Regulated Markets and Multilateral Negotiation Systems of the BME Group. We have selected the Ibex 35 because it is an indicator that shows the markets evolution as a whole, since it represents approximately 90% of the effective trade in the Spanish Stock exchange. The sample period spans from 2nd of January 2015 to the 27th of October 2017. The raw data that has been used to execute the experiments are price ticks from Ibex 35 for each of the trading sessions sampled every 3 seconds starting from 9.00 a.m to the closure of the session at 17:30 p.m. After some pre-processing to remove missing data, the resulting database consists of 724 time series, one per trading session, of length 961 which consist of measurements aggregated to every 30 seconds. The first 506 sessions (70%) are used for training the early classification framework, whereas the rest are used for testing (30%). Then, the class variable is defined in two different ways:

- Binary class: -1 if the price has decreased and 1 if it has increased over a trading session.

- Three-category class: -1 if the price has decreased more than a 0.5% of the initial daily price, 1 if it has increased more than a 0.5% and, 0 in the rest of the cases.

As in the previous sections, the $(h_1, h_2, ..., h_L)$ classifiers and built only every 5% of the length of the series, in order to reduce computational burden, and the rest of the parameters of the multi-objective early classification framework are also left unchanged. We train the probabilistic classifiers using the training set and run the NSGA II optimization algorithm to obtain the set of non-dominated trigger functions for this specific problem. Then, these solutions are applied to the testing set and the early class-predictions are obtained and evaluated. In Figure 10 we represent the obtained results [7]:



(a) 2-class problem

(b) 3-class problem

Figure 10: Solutions obtained by our proposal for the Ibex35 database.

To begin with the interpretation of the results, we must say that in both the 2-class and the 3-class problems, the class-frequencies of the training and testing datasets are balanced, and also the class-accuracies obtained by the method. As can be seen in the figure, in the binary-class case, the accuracy results vary between 54% and a bit over 60%, whereas the earliness goes from around 5% to a bit over 50%. Note that all the accuracies obtained are higher than those that would be obtained by a random classifier, and they are all reached at the latest at midday, when only half of the series has been seen on average. In the case of the three-class problem, the accuracy results vary from around 45% to almost 50%, and the earliness results are even better that in the previous case, with a maximum of around 26%. As in the previous case, the accuracy results are better than those that would be obtained by the random classifier, and these are obtained very early in the day.

The economic implications of the obtained results are also significant. Firstly, they confirm the predictability of the movement of the index Ibex 35 and its return on intraday momentum. In general terms we can point out that prediction accuracy increases when the available information is more abundant, as expected. This is because the longer we wait, the more information we have and the higher resistance of our prediction to noise [23, 31]. Secondly, the return predictabil-

[7] Recall, that the plotted points are non necessarily non-dominated because the solutions have been obtained by using the training set but applied and evaluated using the independent testing set

ity is particularly important in the shorter forecast horizon. The shorter the time interval, the more times a trading strategy can be applied to take advantage and so the greater the potential for high annualized returns [33]. Thirdly, and most significantly, an optimal combination of the three key aspects to implement cost-effective trading strategies is achieved with our early classification framework: speed, accuracy and simplicity in terms of the number of inputs required. Despite using only univariate time series (only the actual stock prices) and a simple linear stopping rule combined with a set of classifiers, results are always higher than those that would be obtained by random classifiers and, depending on the time of day, we can obtain up to 60% of accuracy in the two-class problem.

We also generate early signals for each trading session so that you can have an informative advantage. Traders continuously pursue these advantages even if it is of short duration, as it can generate higher returns [17].

Finally, the early classification framework allows designing different strategies according to investors' risk aversion. Thus, a very risk-averse investor will choose a high hit probability and can expect to wait for more time, while a less risk-averse investor will choose a lower probability of success but take advantage of taking positions faster.

7. Conclusions and Future Work

Early classification is a supervised learning task which can be very naturally interpreted and formulated as a multi-objective optimization problem. In this paper, we have formally defined this problem as such for the first time, and we have proposed a novel framework, which deals with the problem using multi-objective optimization methods.

The advantages of this method are quite evident. In only one execution of the algorithm, we obtain a complete set of solutions which balances the two objectives, earliness and accuracy, in different possible ways. The user can then select the solution most suited to its needs without having to tune or select any specific parameters in advance, as in other previous early classification methods.

We have evaluated our method using 45 benchmark datasets, obtaining superior results in terms of dominance and other multi-objective evaluation measures, when comparing to other state-of-the-art solutions. Additionally, we have presented a real case study from the financial domain, which shows the applicability of the proposed solution.

As an interesting future research line, we propose automatically learning the shape of the stopping rule using genetic programming algorithms. Additionally, we would also like to work further on the case-study in the financial area, adapting the methodology to this specific problem, for example, considering other common prediction horizons, or designing cost functions more adapted to the problem at hand.

Acknowledgements

We would like to thank the UCR archive for providing access to the data used in this study. This work has been partially supported by the Research Groups 2013-2018 (IT-609-13) programs (Basque Government), TIN2016-78365-R (Spanish Ministry of Economy, Industry and Competitiveness) and Severo Ochoa Program SEV-2013-0323 (Spanish Ministry of Economy, Industry and Competitiveness).

References

[1] A. Abanda, U. Mori, J.A. Lozano, A review on distance based time series classification, https://arxiv.org/abs/1806.04509.

[2] A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh, The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances, Data Min. Knowl. Discov. 31 (3) (2017) 606–660.

[3] J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market, Journal of Computational Science 2 (1) (2011) 1 – 8.

[4] A. Bregón, M. A. Simón, J. J. Rodríguez, C. Alonso, B. Pulido, I. Moro, Early Fault Classification in Dynamic Systems Using Case-Based Reasoning, in: Proceeding CAEPIA'05 Proceedings of the 11th Spanish association conference on Current Topics in Artificial Intelligence, 2006, pp. 211–220.

[5] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The ucr time series classification archive, `www.cs.ucr.edu/~eamonn/time_series_data/` (July 2015).

[6] A. Dachraoui, A. Bondu, Early Classification of Time Series as a Non Myopic Sequential Decision Making Problem, in: ECML PKDD 2015, Vol. Part I, 2015, pp. 433–447. doi:10.1007/978-3-319-23528-8.

[7] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi:10.1109/4235.996017.

[8] G. J. Deboeck, Training on the Edges: Neural, Genetic and Fuzzy Systems for Chaotic Financial Marketing, wiley Edition, New York, 1994.

[9] P. Esling, C. Agon, Time-series data mining, ACM Computing Surveys (CSUR) 45 (1) (2012) 1–34.

[10] R. S. Evans, K. G. Kuttler, K. J. Simpson, S. Howe, P. F. Crossno, K. V. Johnson, M. N. Schreiner, J. F. Lloyd, W. H. Tettelbach, R. K. Keddington, A. Tanner, C. Wilde, T. P. Clemmer, Automated detection of physiologic deterioration in hospitalized patients., Journal of the American Medical Informatics Association : JAMIA 22 (2) (2015) 350–60. doi:10.1136/amiajnl-2014-002816.

[11] E. F. Fama, Efficient capital markets: A review of theory and empirical work, The Journal of Finance 25 (1970) 383417.

[12] T.-C. Fu, A Review on Time Series Data Mining, Engineering Applications of Artificial Intelligence 24 (1) (2011) 164–181.

[13] M. F. Ghalwash, D. Ramljak, Z. Obradovic, Early classification of multivariate time series using a hybrid HMM/SVM model, in: IEEE International Conference on Bioinformatics and Biomedicine, 2012, pp. 1–6.

[14] M. F. Ghalwash, V. Radosavljevic, Z. Obradovic, 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, New York, New York, USA, 2014, pp. 402–411. doi:10.1145/2623330.2623694.

[15] N. Hatami, C. Chira, Classifiers With a Reject Option for Early Time-Series Classification, in: IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), 2013, pp. 9–16.

[16] G. He, Y. Duan, R. Peng, X. Jing, T. Qian, L. Wang, Early classification on multivariate time series, Neurocomputing 149 (2015) 777–787. doi:10.1016/j.neucom.2014.07.056.

[17] O. Kadan, Trading in the Presence of Short-Lived Private Information : Evidence from Analyst Recommendation Changes Trading in the Presence of Short-Lived Private Information, Journal of Financial and Quantitative Analysis (JFQA), Forthcoming.

[18] L.-J. Kao, C.-C. Chiu, C.-J. Lu, C.-H. Chang, A hybrid approach by integrating wavelet-based feature extraction with mars and svr for stock index forecasting, Decision Support Systems 54 (3) (2013) 1228 – 1244.

[19] R. J. Kate, Using dynamic time warping distances as features for improved time series classification, Data Mining and Knowledge Discoverydoi:10.1007/s10618-015-0418-x.

[20] Y. Kara, M. A. Boyacioglu, mer Kaan Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange, Expert Systems with Applications 38 (5) (2011) 5311 – 5319.

[21] N. Lama, M. Girolami, vbmp: Variational Bayesian Multinomial Probit Regression, R package version 1.46.0 (2017).

[22] M. T. Leung, H. Daouk, A.-S. Chen, Forecasting stock indices: a comparison of classification and level estimation models, International Journal of Forecasting 16 (2) (2000) 173 – 190.

[23] C.-C. Lin, C.-S. Chen, A.-P. Chen, Using intelligent computing and data stream mining for behavioral finance associated with market profile and financial physics, Applied Soft Computing.

[24] B. G. Malkiel, The efficient market hypothesis and its critics, Journal of Economic Perspectives 17 (2003) 59–82.

[25] O. Mersmann, mco: Multiple Criteria Optimization Algorithms and Related Functions, r package version 1.0-15.1 (2014).

[26] U. Mori, A. Mendiburu, E. Keogh, J. A. Lozano, Reliable early classification of time series based on discriminating the classes over time, Data Mining and Knowledge Discovery (2016) 1–31doi:10.1007/s10618-016-0462-1.

[27] U. Mori, A. Mendiburu, S. Dasgupta, J. A. Lozano, Early Classification of Time Series by Simultaneously Optimizing the Accuracy and Earliness,

IEEE Transactions on Neural Networks and Learning Systems In press (2017) 1–10. doi:10.1109/TNNLS.2017.2764939.

[28] N. Parrish, H. S. Anderson, D. Y. Hsiao, Classifying With Confidence From Incomplete Information, Journal of Machine Learning Research 14 (2013) 3561–3589.

[29] H. Sakoe, S. Chiba, DynaMic Programming Algorithm Optimization for Spoken WordRecognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 26 (1) (1978) 43–49.

[30] L. Scrucca, GA: A package for genetic algorithms in R, Journal of Statistical Software 53 (4) (2013) 1–37.

[31] S. Shen, H. Jiang, T. Zhang, Stock market forecasting using machine learning algorithms, Tech. rep., Department of Electrical Engineering, Stanford University (2012).

[32] R. Tavenard, S. Malinowski, Cost-aware early classification of time series, in: P. Frasconi, N. Landwehr, G. Manco, J. Vreeken (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2016, pp. 632–647.

[33] A. Timmermann, Elusive return predictability, International Journal of Forecasting 24 (1) (2008) 1 – 18.

[34] J.-Z. Wang, J.-J. Wang, Z.-G. Zhang, S.-P. Guo, Forecasting stock indices with back propagation neural network, Expert Systems with Applications 38 (11) (2011) 14346 – 14355.

[35] B. Weng, Application of machine learning techniques for stock market prediction. short-term stock movement and price., Ph.D. thesis, Auburn University (2017).

[36] Z. Xing, J. Pei, P. S. Yu, Early classification on time series, Knowledge and Information Systems 31 (1) (2011) 105–127.

[37] Z. Xing, P. S. Yu, K. Wang, Extracting Interpretable Features for Early Classification on Time Series, in: Proceedings of the Eleventh {SIAM} International Conference on Data Mining, 2011, pp. 247–258.

[38] L. Ye, E. Keogh, Time Series Shapelets : A New Primitive for Data Mining, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 947–956.