Knowledge Extraction with Interval Temporal Logic Decision Trees

Guido Sciavicco

Department of Mathematics and Computer Science, University of Ferrara, Italy
guido.sciavicco@unife.it

Ionel Eduard Stan

Department of Mathematics and Computer Science, University of Ferrara, Italy
Department of Mathematical, Physical, and Computer Sciences, University of Parma, Italy
ioneleduard.stan@unife.it

Abstract

Multivariate temporal, or time, series classification is, in a way, the temporal generalization of (numeric) classification, as every instance is described by multiple time series instead of multiple values. Symbolic classification is the machine learning strategy to extract explicit knowledge from a data set, and the problem of symbolic classification of multivariate temporal series requires the design, implementation, and test of ad-hoc machine learning algorithms, such as, for example, algorithms for the extraction of temporal versions of decision trees. One of the most well-known algorithms for decision tree extraction from categorical data is Quinlan's ID3, which was later extended to deal with numerical attributes, resulting in an algorithm known as C4.5, and implemented in many open-sources data mining libraries, including the so-called Weka, which features an implementation of C4.5 called J48. ID3 was recently generalized to deal with temporal data in form of timelines, which can be seen as discrete (categorical) versions of multivariate time series, and such a generalization, based on the interval temporal logic HS, is known as Temporal ID3. In this paper we introduce Temporal C4.5, that allows the extraction of temporal decision trees from undiscretized multivariate time series, describe its implementation, called Temporal J48, and discuss the outcome of a set of experiments with the latter on a collection of public data sets, comparing the results with those obtained by other, classical, multivariate time series classification methods.

2012 ACM Subject Classification Theory of computation; Theory of computation → Logic

Keywords and phrases Interval Temporal Logic, Decision Trees, Explainable AI, Time series

Digital Object Identifier 10.4230/LIPIcs.TIME.2020.9

Acknowledgements Computational resources have been offered by the University of Udine, Italy, supported by the PRID project *Efforts in the uNderstanding of Complex interActing SystEms* (ENCASE) and the authors acknowledge the partial support by the Italian INDAM GNCS project *Strategic Reasoning and Automated Synthesis of Multi-Agent Systems*.

1 Introduction

A *labelled data set* $\mathcal{D} = \{D_1, \ldots, D_m\}$ is a set of instances described by a set of numerical and/or categorical attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$ and associated to a set of classes $\mathcal{C} = \{C_1, \ldots, C_q\}$. *Supervised symbolic classification* is the machine learning strategy for extracting an explicit (logical) theory that describes a labelled data set $\mathcal{D}$. It is usually opposed to *supervised functional classification*, which includes linear regression, logistic regression, neural networks, and many other black-box and function-extraction mechanisms. There are many symbolic classification methods, which can be broadly distinguished into *tree-based* and *rule-based*. Tree-based classification models can be *single* or *multiple*; multiple tree-based models, however, while still symbolic in nature, are not interpretable in the same sense as single trees are. Tree-based classification models are also known as *decision trees*, and they

27th International Symposium on Temporal Representation and Reasoning (TIME 2020).
Editors: Emilio Muñoz-Velasco, Ana Ozaki, and Martin Theobald; Article No. 9; pp. 9:1–9:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

can be described in a very abstract (inductive) way: a decision tree is a class, or it is a k-ways decision followed by k decision trees. The introduction of decision trees can be dated back to [10, 11]. The problem of extracting the *optimal* decision tree from a data set is NP-hard [9], which justifies the use of sub-optimal approaches. The *ID3* algorithm [10] is a greedy approach to decision tree extraction; it is based on a simple concept: given a labelled data set $\mathcal{D}$, one takes a *decision* by choosing the attribute A_i and the value(s) on the domain A_i that return(s) the highest information, obtaining, in general, a k-ways partition of $\mathcal{D}$ in $\mathcal{D}_1, \ldots, \mathcal{D}_k$. The information is measured in terms of the classes that occur in $\mathcal{D}$ and in $\mathcal{D}_1, \ldots, \mathcal{D}_k$. Each decision can be expressed as a propositional letter; since alternative branches can be seen as logical (exclusive) disjunctions, and successive decisions on the same branches can be seen as logical conjunctions, a decision tree as a whole can be seen as a set of propositional formulas. In other words, a decision tree is a propositional description of the data set on which it is learned.

A *time series* is a set of variables that change over time, and they can be *univariate* or *multivariate*. Each variable of a multivariate time series is an ordered collection of N real values, instead of a single value. So, a *labelled temporal data set* $\mathcal{T} = \{T_1, \ldots, T_m\}$ is a set of temporal instances described by a set of temporal attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$, each being a N-points time series, and associated to a set of classes $\mathcal{C} = \{C_1, \ldots, C_q\}$. Multivariate time (or temporal) series emerge in many application contexts. The temporal history of some hospitalized patient can be described by the time series of the values of his/her temperature, blood pressure, and oxygenation; the pronunciation of a word in sign language can be described by the time series of the relative and absolute positions of the ten fingers w.r.t. some reference point; different sport activities can be distinguished by the time series of some relevant physical quantities. In the current literature, time series classification algorithms can be instance-based, feature-based, and timeline-based. *Instance-based* methods are essentially all built on the notion of distance between two time series, by means of which a time series can be classified using, e.g., the *Nearest Neighbor* (NN) algorithm. The most widely accepted notions of distances are the *Euclidean Distance* (ED) and the *Dynamic Time Warping* (DTW) [14]. Intuitively, ED is a one-to-one alignment method, while DTW is one-to-many, as it allows one to compare time series even of different scales. Such methods have been systematically applied to a variety of multivariate time series data sets in [2] (the univariate case is dealt with by the same authors in [3]). *Feature-based* methods, on the other hand, consist of flattening the time series, and describe each one of them via a set of values (e.g., mean, variance, maximum, minimum). These descriptions, in turn, can be used as the input of a static learning algorithm. Feature-based techniques are widespread in the data science community, because they are conceptually simple, and allow one to use familiar learning methods; unfortunately, the theories obtained in this way are not always interpretable, and the quality of the models in term of performances is not always acceptable. An extensive comparison between instance-based and feature-based methods can be found in [7], in which the authors also present an algorithm that allows one to automatically choose the best features to represent a time series data set for it to be classified. Finally, A *timeline* can be considered as the discretized version of a multivariate time series. For each single variable, one produces a set of propositional letters that describe the values of that variable, or its (first, second, ...) derivative, on every possible interval of time, and then describes a multivariate time series on a single line by joining the propositional, interval description of all variables. A general method to translate a multivariate time series into a timeline is described in [13], and *Temporal ID3* [5] can be considered an example of *timeline-based* classification of multivariate time series, that uses the interval temporal logic HS [8] to describe a decision tree.

HS	Allen's relations
$\langle A\rangle$	$[x,y]R_A[x',y'] \Leftrightarrow y = x'$
$\langle L\rangle$	$[x,y]R_L[x',y'] \Leftrightarrow y < x'$
$\langle B\rangle$	$[x,y]R_B[x',y'] \Leftrightarrow x = x', y' < y$
$\langle E\rangle$	$[x,y]R_E[x',y'] \Leftrightarrow y = y', x < x'$
$\langle D\rangle$	$[x,y]R_D[x',y'] \Leftrightarrow x < x', y' < y$
$\langle O\rangle$	$[x,y]R_O[x',y'] \Leftrightarrow x < x' < y < y'$

Graphical representation

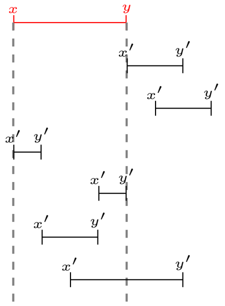

Figure 1 Allen's interval relations and HS modalities.

In this paper we design a decision tree learning algorithm, *Temporal C4.5*, that generalizes Temporal ID3 in the same way in which the algorithm C4.5 [12] generalizes ID3, that is, by introducing the possibility of learning from continuous attributes. In this case, however, (non-discretized) time series are described *only* by continuous (time-changing) values, so, in the current version, Temporal C4.5 does not admit categorical attributes. We consider one of the most representative implementations of C4.5, called *J48*, available in the Weka open-source learning suite [15], and we modify it by introducing decisions based on the interval temporal logic HS. The main contributions of this paper are:

(i) the definition of a general theory of decision trees, which is used to guide our generalization from the static to the temporal case;

(ii) the first implementation of a symbolic classification algorithm for time series that deals with the raw data (i.e., without applying any pre-abstraction method), whose extracted theory is expressed in the interval temporal logic HS;

(iii) a comparison of the performances of our implementation against existing methods on the public data used in [2].

2 Preliminaries

Classification of multivariate temporal series. A *time series* is a set of variables that change over time, and they can be *univariate* or *multivariate*. Each variable (or *channel*) of a multivariate time series is an ordered collection of N real values, instead of a single value, so that a single time series can be described as follows:

$$T = \begin{cases} A_1 & = & a_{1,1}, & a_{1,2}, & \ldots, & a_{1,N} \\ A_2 & = & a_{2,1}, & a_{2,2}, & \ldots, & a_{2,N} \\ \ldots & & \ldots & & & \\ A_n & = & a_{n,1}, & a_{n,2}, & \ldots, & a_{n,N}. \end{cases} \tag{1}$$

So, a *labelled temporal data set* $\mathcal{T} = \{T_1, \ldots, T_m\}$ is a set of temporal instances described by a set of temporal attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$, each being a N-points time series, and associated to a set of classes $\mathcal{C} = \{C_1, \ldots, C_q\}$. A temporal data set can be viewed as a $m \times n$ matrix where the ith row, $1 \leq i \leq m$, is a multivariate time series and the jth column, $1 \leq j \leq n$, is an attribute. The *multivariate time series supervised classification problem* is the problem of finding a formula (symbolic classification) or a function (functional classification) that associates multivariate time series to classes.

Timelines and interval temporal logic. A multivariate time series can be discretized without eliminating the temporal component. In [13] the authors introduce the notion of *timeline* and present a procedure that transforms multivariate time series into timelines. Time series describe continuous processes; when discretized, it makes little sense to model their values at each point, but, instead, they are naturally represented in a interval-based ontology. Thus, if a static numerical data set is naturally represented in propositional logic, a multivariate time series is naturally represented in an interval temporal logic.

Let $[N]$ an initial subset of $\mathbb{N}$ of length N. An *interval* over $[N]$ is an ordered pair $[x, y]$, where $x, y \in [N]$ and $x < y$, and we denote by $\mathbb{I}([N])$ the set of all intervals over $[N]$. If we exclude the identity relation, there are 12 different Allen's relations between two intervals in a linear order [1]: the six relations R_A (adjacent to), R_L (later than), R_B (begins), R_E (ends), R_D (during), and R_O (overlaps), depicted in Figure 1, and their inverses, that is, $R_{\bar{X}} = (R_X)^{-1}$, for each $X \in \mathcal{X}$, where $\mathcal{X} = \{A, L, B, E, D, O\}$. Halpern and Shoham's modal logic of temporal intervals (HS) is defined from a set of propositional letters $\mathcal{AP}$, and by associating a universal modality $[X]$ and an existential one $\langle X \rangle$ to each Allen's relation R_X. Formulas of HS are obtained by

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle\varphi \mid \langle \bar{X} \rangle\varphi, \tag{2}$$

where $p \in \mathcal{AP}$ and $X \in \mathcal{X}$. The other Boolean connectives and the logical constants, e.g., $\rightarrow$ and $\top$, as well as the universal modalities $[X]$, can be defined in the standard way, i.e., $[X]p \equiv \neg\langle X \rangle\neg p$. For each $X \in \mathcal{X}$, the modality $\langle \bar{X} \rangle$ (corresponding to the inverse relation $R_{\bar{X}}$ of R_X) is said to be the *transpose* of the modalities $\langle X \rangle$, and vice versa. The semantics of HS formulas is given in terms of *timelines* $T = \langle \mathbb{I}([N]), V \rangle^1$, where $V : \mathcal{AP} \rightarrow 2^{\mathbb{I}([N])}$ is a *valuation function* which assigns to each atomic proposition $p \in \mathcal{AP}$ the set of intervals $V(p)$ on which p holds. The *truth* of a formula φ on a given interval $[x, y]$ in an interval model T is defined by structural induction on formulas as follows:

$$\begin{array}{lll} T, [x, y] \Vdash p & \text{if} & [x, y] \in V(p), \text{ for } p \in \mathcal{AP}; \\ T, [x, y] \Vdash \neg\psi & \text{if} & T, [x, y] \not\Vdash \psi; \\ T, [x, y] \Vdash \psi \vee \xi & \text{if} & T, [x, y] \Vdash \psi \text{ or } T, [x, y] \Vdash \xi; \\ T, [x, y] \Vdash \langle X \rangle\psi & \text{if} & \text{there is } [w, z] \text{ s.t } [x, y]R_X[w, z] \text{ and } T, [w, z] \Vdash \psi; \\ T, [x, y] \Vdash \langle \bar{X} \rangle\psi & \text{if} & \text{there is } [w, z] \text{ s.t } [x, y]R_{\bar{X}}[w, z] \text{ and } T, [w, z] \Vdash \psi. \end{array} \tag{3}$$

A time series can be seen as a timeline based on a suitable propositional signature. As for example, consider a time series that records medical values of some hospitalized patient: temperature, blood pressure, and oxygenation, as in Figure 2, top. The information can be adequately subsumed into a timeline such as in Figure 2, bottom, provided that a suitable propositional signature is given. In our example, we (arbitrarily) decided that, for instance, the value 38 is an informative splitting point, and a propositional letter $(Te \leq 38)$ can be created. In [13], a methodology that allows one to perform such discretization is presented, and in [5], a temporal decision tree extraction method (Temporal ID3), that takes a temporal data set in form of timelines and extracts a decision tree whose edges are labelled with decisions written in HS, is studied. The main limitation of such an approach is that the discretization method does not take into account the predictive capabilities of the decisions (that is, of the propositional letters), because it is run off-line, so to say: in machine learning terms, it is a *filter*. Temporal C4.5 is aimed to close precisely this gap, allowing the propositional signature to emerge *during* the decision tree extraction, not before, precisely as C4.5 does on non-temporal data.

[1] We deliberately use the symbol T to indicate both a timeline and a time series.

time	O_2-sat (O_2)	art. press. (Pr)	temp (Te)
1	88	105	37
2	89	107	37
3	90	110	39
4	85	108	39
5	82	102	37

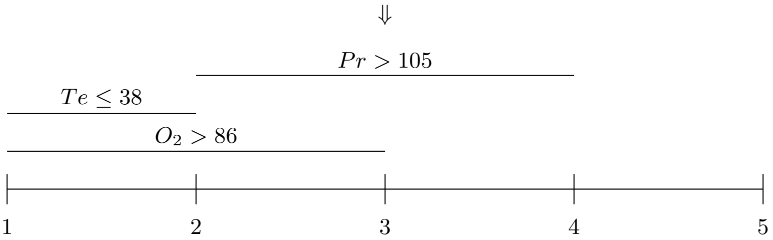

Figure 2 Using timelines to discretize multivariate time series: an example with a single time series.

3 A Theory of Decision Trees

A theory of static decision trees. Decision trees are a very well-known construct. While in the literature there has been a great effort to improve their implementation, versatility, and applicability, a formal definition of the structure and semantics of decision trees is necessary to correctly define their temporal generalization. In this section, for the sake of simplicity of explanation, we restrict our attention to the case of binary decision trees for binary classification, both in the static and the temporal case. Generalizing our approach to the case of k-ary trees and multiple classes is immediate.

Consider a labelled static data set $\mathcal{D} = \{D_1, \ldots, D_m\}$ described by the set of attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$ and associated to the classes $\mathcal{C} = \{Yes, No\}$. We denote the *domain* of an attribute A by $dom(A)$. The language of static decision trees encompasses a set $\mathcal{S}$ of *propositional decisions*:

$$\mathcal{S} = \{A \bowtie a \mid A \in \mathcal{A}, a \in dom(A)\}, \tag{4}$$

where $\bowtie \in \{\leq, =\}$. Binary static decision trees are formulas of the following grammar:

$$\hat{\varphi} ::= (S \wedge \hat{\varphi}) \veebar (\neg S \wedge \hat{\varphi}) \mid C. \tag{5}$$

where $C \in \mathcal{C}$ and $S \in \mathcal{S}$. The symbol $\veebar$ indicates the exclusive *or*, while the symbol $\wedge$ indicates the classical propositional *and*. Every non-leaf node of a decision tree has two children and every edge is decorated with a decision. Leaves are decorated with a class. A decision S is interpreted over a single instance D using classical propositional logic. We say that D *satisfies* the decision $A \leq a$ (resp., $A = a$) if the attribute A has a value less than or equal to (resp., equal to) $a \in dom(A)$ in D, and we use the symbol $D \models (A \leq a)$ (resp., $D \models (A < a)$).

A decision tree is interpreted over a labelled data set $\mathcal{D}$ via the semantic relation $\hat{\models}_\theta$, which generalizes $\models$ from single instances to data sets: we need to define the notion of a data set satisfying $\hat{\varphi}$ with parameter θ, that is, $\mathcal{D} \hat{\models}_\theta \hat{\varphi}$. Unlike the classical propositional logic, in this case the truth relation is parametric; the parameter θ formalizes the notion of how well a decision tree $\hat{\varphi}$ represents $\mathcal{D}$. Let D an instance of $\mathcal{D}$. We denote by $C(D)$ the true class of D, and by $\hat{\varphi}(D)$ the class that $\hat{\varphi}$ predicts for D. The generic performance of $\hat{\varphi}$ on $\mathcal{D}$ can be measured in terms of its *confusion matrix*, which, for each given instance, expresses one of four possible, mutually exclusive, indicators (*true positive*, *true negative*, *false positive*, and *false negative*) by comparing $C(D)$ and $\hat{\varphi}(D)$:

$$\begin{array}{r|c|c|}
 & C(D) = No & C(D) = Yes \\ \cline{2-3}
\hat{\varphi}(D) = No & \text{True Negative (TN)} & \text{False Negative (FN)} \\ \cline{2-3}
\hat{\varphi}(D) = Yes & \text{False Positive (FP)} & \text{True Positive (TP)} \\ \cline{2-3}
\end{array} \tag{6}$$

The root of a tree $\hat{\varphi}$ is associated with the data set $\mathcal{D}$ on which it is interpreted, and, in general, each node of the tree is associated with a subset $\mathcal{D}' \subset \mathcal{D}$ and a binary decision S. A set $\mathcal{D}'$ is partitioned into two subsets $\mathcal{D}'_1$ and $\mathcal{D}'_2$, that contain, respectively the instances that satisfy S and those that do not. The subset of $\mathcal{D}$ associated with a leaf is also labelled with a class C, meaning that every instance in it is classified with C, generating a certain amount of misclassifications. From the leaves, one can inductively compute the confusion matrix of each node. The confusion matrix of the root is the one we associate with the tree itself. The rules for $\hat{\models}_\theta$ are now immediate:

$$\begin{array}{lll}
\mathcal{D} \hat{\models}_\theta No & \text{if} & \theta = \begin{array}{|c|c|}\hline |\mathcal{D}_{No}| & |\mathcal{D}| - |\mathcal{D}_{No}| \\ \hline 0 & 0 \\ \hline \end{array}\text{, where} \\
 & & \mathcal{D}_{No} = \{D \in \mathcal{D} \mid C(D) = No\}, \\
\mathcal{D} \hat{\models}_\theta Yes & \text{if} & \theta = \begin{array}{|c|c|}\hline 0 & 0 \\ \hline |\mathcal{D}| - |\mathcal{D}_{Yes}| & |\mathcal{D}_{Yes}| \\ \hline \end{array}\text{, where} \\
 & & \mathcal{D}_{Yes} = \{D \in \mathcal{D} \mid C(D) = Yes\}, \\
\mathcal{D} \hat{\models}_\theta (S \wedge \hat{\varphi}_1) \forall (\neg S \wedge \hat{\varphi}_2) & \text{if} & \theta = \theta_1 + \theta_2, \mathcal{D}_1 \hat{\models}_{\theta_1} \hat{\varphi}_1, \text{ and } \mathcal{D}_2 \hat{\models}_{\theta_2} \hat{\varphi}_2, \text{ where} \\
 & & \mathcal{D}_1 = \{D \in \mathcal{D} \mid D \models S\}, \mathcal{D}_2 = \{D \in \mathcal{D} \mid D \models \neg S\}, \\
 & & \mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2, \text{ and } \mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset.
\end{array} \tag{7}$$

Observe that computing the confusion matrix on a node generalizes every classical notion of performance, such as accuracy, precision, recall, among others.

A theory of temporal decision trees. On the basis of the above notions, we can now define the concept of temporal decision tree.

Let us now consider a labelled temporal data set $\mathcal{T} = \{T_1, \ldots, T_n\}$, in which every instance is described by the time series $\mathcal{A} = \{A_1, \ldots, A_n\}$ (recall that each attribute is a univariate time series, in this case) and, as before, classified in one of two classes from the set $\mathcal{C} = \{Yes, No\}$. We assume that all attributes have the same temporal length, N. Because time series describe continuous processes, our decisions will be taken on the value of a single channel over an interval of time, and we describe such decisions using propositional interval temporal logic. Unlike static attributes, however, temporal attributes can be analyzed over multiple dimensions. Consider an interval of time $[x, y]$ and an attribute A that varies on it. As in the static case we can ask the question $A \bowtie a$ over the entire interval, where $\bowtie \in \{\leq, =\}$, which is positively answered if *every value* of A in the interval $[x, y]$ respects the given constraint. But unlike the static case, we do not ask if $A \bowtie a$ only in the *current interval* but also if *there exists an interval*, related to the current one, in which that holds, so that the decision becomes $\langle X \rangle (A \bowtie a)$. This implies, among other things, that the relation $>$ cannot be defined as the negation of $\leq$: when we apply the negation, indeed, we negate *both* $\langle X \rangle$ and $\bowtie$, which amounts to say that if we want to check if $\langle X \rangle (A > a)$ we have to do it explicitly. Therefore, in this case, $\bowtie \in \{\leq, =, >\}$. Moreover, in order to allow a certain degree of uncertainty, we may relax the requirement $A \bowtie a$ over a given interval $[x, y]$ by asking that *at least a certain fraction* of the values of A in the interval $[x, y]$ meet this condition; we denote this relaxed decision by $A \bowtie_\alpha a$, where $\alpha \in (0, 1]$ is a real parameter. Finally, we need to remember that in certain applications the values may not be as important as the trends, that is, the value of the *discrete derivative* of A at a certain degree. We denote the discrete derivative of A at degree z by A^z (identifying A^0 with A), and, consequently, a generic temporal decision by $\langle X\rangle(A^z \bowtie_\alpha a)$, with $a \in dom(A^z)$. Thus, the language of temporal decision trees encompasses the following set of *temporal decisions*:

$$\begin{aligned}\mathcal{S} &= \{\langle X\rangle(A^z \bowtie_\alpha a), \langle\bar{X}\rangle(A^z \bowtie_\alpha a) \mid X \in \mathcal{X}, A \in \mathcal{A}, a \in dom(A^z)\} \cup \\ &\quad \{A^z \bowtie_\alpha a \mid A \in \mathcal{A}, a \in dom(A^z)\},\end{aligned} \tag{8}$$

where $\mathcal{X} = \{A, L, B, E, D, O\}$ are interval operators of the language of HS. Temporal decision trees are formulas obtained from (5), in which propositional decisions have been replaced by temporal decisions. A temporal decision is interpreted over a single multivariate time series T and interval $[x, y]$, by using the notion of semantic relation $\Vdash$ recalled in Section 2; therefore, we use the notation $T, [x, y] \Vdash \langle X\rangle(A^z \bowtie_\alpha a)$ or $T, [x, y] \Vdash \langle\bar{X}\rangle(A^z \bowtie_\alpha a)$. Formally, given a point t, we denote by $A^z(t)$ the value of the z-th discrete derivative of the attribute A at the point t, and given an interval $[x, y]$, we denote by $[x, y]^{A^z \bowtie a}$ the following set:

$$[x, y]^{A^z \bowtie a} = \{t \mid x \leq t \leq y, A^z(t) \bowtie a\}. \tag{9}$$

Therefore we have:

$$\begin{aligned}T, [x, y] \Vdash \langle X\rangle(A^z \bowtie_\alpha z) \quad \text{if} \quad & \text{there is } [w, z] \text{ s.t } [x, y] R_X [w, z] \text{ and} \\ & |[w, z]^{A^z \bowtie a}| \geq \lceil \alpha \cdot (z - w + 1)\rceil.\end{aligned} \tag{10}$$

We now want to define the notion of a temporal data set satisfying $\hat{\varphi}$ with parameter θ, that is, $\mathcal{T} \hat{\Vdash}_\theta \hat{\varphi}$. In the static case, from a data set $\mathcal{D}$ one computes immediately the two data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ entailed by a decision S. In the temporal case, however, this requires more effort. Every instance of the temporal data set $\mathcal{T}$ associated with the root of $\hat{\varphi}$ is assigned the reference interval $[0, 1]$ by default; observe that, since the language of HS encompasses a set of jointly exhaustive operators, this requirement does not decrease the expressive power of temporal decisions. Given a temporal decision S over $\mathcal{T}$, computing the set $\mathcal{T}_1$ of all instances that do satisfy S implies assigning to each instance $T \in \mathcal{T}_1$ a potentially different reference interval; on the contrary, computing $\mathcal{T}_2$ implies leaving the reference interval of its members unchanged. So, for example, given $\mathcal{T}$, in which every instance T is assigned a reference interval $[x_T, y_T]$, and given the decision $S = \langle A\rangle(A \leq a)$, we say that:

$$\begin{aligned}\mathcal{T}_1 &= \{T \in \mathcal{T} \mid \exists [y_T, z_T](y_T < z_T \wedge T, [y_T, z_T] \Vdash (A \leq a))\} \\ \mathcal{T}_2 &= \{T \in \mathcal{T} \mid \forall [y_T, z_T](y_T < z_T \rightarrow T, [y_T, z_T] \nVdash (A \leq a))\}.\end{aligned} \tag{11}$$

In the particular case in which S is static, the reference interval does not change for $\mathcal{T}_1$, either. For a temporal decision S, we use the notation $T \Vdash S$ (resp., $T \Vdash \neg S$) to identify the members of $\mathcal{T}_1$ (resp., $\mathcal{T}_2$). Observe that S entails unique $\mathcal{T}_1$ and $\mathcal{T}_2$, but not unique (new) reference intervals for the members of $\mathcal{T}_1$; however, this choice is implementative, not theoretical. At this point, the notion of how well a temporal data set $\mathcal{T}$ satisfies a decision tree $\hat{\varphi}$ becomes immediate, and completely equivalent to the static case:

$$\begin{array}{lll}
\mathcal{T} \hat{\Vdash}_{\theta} No & \text{if} & \theta = \begin{array}{|c|c|}\hline |\mathcal{T}_{No}| & |\mathcal{T}| - |\mathcal{T}_{No}| \\ \hline 0 & 0 \\ \hline \end{array}, \text{ where} \\
 & & \mathcal{T}_{No} = \{T \in \mathcal{T} \mid C(T) = No\}, \\
\mathcal{T} \hat{\Vdash}_{\theta} Yes & \text{if} & \theta = \begin{array}{|c|c|}\hline 0 & 0 \\ \hline |\mathcal{T}| - |\mathcal{T}_{Yes}| & |\mathcal{T}_{Yes}| \\ \hline \end{array}, \text{ where} \\
 & & \mathcal{T}_{Yes} = \{T \in \mathcal{T} \mid C(T) = Yes\}, \\
\mathcal{T} \hat{\Vdash}_{\theta} (S \wedge \hat{\varphi}_1) \forall (\neg S \wedge \hat{\varphi}_2) & \text{if} & \theta = \theta_1 + \theta_2, \mathcal{T}_1 \hat{\Vdash}_{\theta_1} \hat{\varphi}_1, \text{ and } \mathcal{T}_2 \hat{\Vdash}_{\theta_2} \hat{\varphi}_2, \text{ where} \\
 & & \mathcal{T}_1 = \{T \in \mathcal{T} \mid T \Vdash S\}, \mathcal{T}_2 = \{T \in \mathcal{T} \mid T \Vdash \neg S\}, \\
 & & \mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2, \text{ and } \mathcal{T}_1 \cap \mathcal{T}_2 = \emptyset.
\end{array} \tag{12}$$

Analogously to the static case, $C(T)$ indicates the true class of the multivariate time series T, and $\hat{\varphi}(T)$ indicates the class assigned by $\hat{\varphi}$.

4 Temporal J48

Entropy-based learning. In [9], it has been proved that computing the optimal decision tree is an NP-hard problem, where the notion of optimality is expressed as the relation between the height and the performance of the tree. In the perspective of practical applications of decision trees to real-life data, this justifies the use of algorithms that return sub-optimal trees, such as ID3 [10]. ID3 is designed for static, categorical data, at it encompasses k-ary splits, but, without loss of generality, we focus our attention on binary splits only.

ID3 is able to learn a tree from a purely categorical data set (i.e., $\bowtie \in \{=\}$ in Equation (4)), and it uses the concepts of information gain and entropy to select the best decision at every node. By identifying frequencies with probabilities, one defines the *information conveyed* by $\mathcal{D}$ (or *entropy* of $\mathcal{D}$) by first measuring the amount of instances in $\mathcal{D}$ that belong to each class C_i (let us denote this subset with $\mathcal{D}_{C_i}$), and, then, by computing:

$$Info(\mathcal{D}) = -\sum_{i=1}^{i=2} (\frac{|\mathcal{D}_{C_i}|}{|\mathcal{D}|} \log(\frac{|\mathcal{D}_{C_i}|}{|\mathcal{D}|})). \tag{13}$$

Intuitively, the entropy is inversely proportional to the purity degree of $\mathcal{D}$ with respect to the class values. *Splitting*, which is the main *greedy* operation in learning a decision tree with ID3, is performed over a specific attribute A. When restricted to categorical attributes and binary splits, a split depends on a particular attribute A and a particular value $a \in dom(A)$, which entail two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$; the former (resp. the latter) contains all those instances D such that $D \models (A = a)$ (resp., $D \models (A \neq a)$ - see Equation (7)). Thus, we can compute the *splitting information* of the pair A, a as follows:

$$InfoSplit(A, a, \mathcal{D}) = \sum_{i=1}^{i=2} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} Info(\mathcal{D}_i), \tag{14}$$

which implies that the *entropy of attribute* A is defined as:

$$InfoAtt(A, \mathcal{D}) = \min_{a \in dom(A)} \{InfoSplit(A, a, \mathcal{D})\}, \tag{15}$$

and, finally, that the *information gain* of A is defined as:

$$Gain(A, \mathcal{D}) = Info(\mathcal{D}) - InfoAtt(A, \mathcal{D}). \tag{16}$$

The algorithm ID3 is based on the idea of recursively splitting the data set over the attribute and the value of its domain that guarantee the greatest information gain, until a certain stopping criterion is met. When non-binary splits are allowed, the concept of splitting information must be slightly modified, but the underlying ideas remain. Given a temporal data set $\mathcal{T}$, one can use the abstraction algorithm presented in [13] to obtain a discretized version of $\mathcal{T}$, in which every mutivariate time series becomes a timeline (as explained in Section 2). The algorithm *Temporal ID3* [5] is able to extract a temporal decision tree that follows the general theory explained in the previous section using the same principles of entropy and information gain. As time series are represented in the form of timelines, Temporal ID3 takes decisions of the type $\langle X\rangle(A = a)$, where A is a discretized attribute and a is one of the possible propositions that emerged from the discretization, and establishes the interval relation, attribute, and propositional letter over which the split is performed according to the entropy principle. In other words, we have:

$$InfoSplit(A, X, a, \mathcal{T}) = \sum_{i=1}^{i=2} \frac{|\mathcal{T}_i|}{|\mathcal{D}|} Info(\mathcal{T}_i), \tag{17}$$

where X takes values in $\mathcal{X} \cup \{eq\}$, eq being the interval temporal relation that captures the current interval only, and T_1, T_2 are computed as explained in the previous section (with $\alpha = 1$ and $z = 0$), and:

$$InfoAtt(A, \mathcal{T}) = \min_{X\in\mathcal{X}\cup\{eq\}, a\in dom(A)} \{InfoSplit(A, X, a, \mathcal{T})\}. \tag{18}$$

The algorithm C4.5 [11] is designed to allow ID3 to cope with numerical data. C4.5 uses exactly the same principles of entropy and information gain introduced for ID3. The main difference, in the static case, lies in allowing $\bowtie$ to take values in $\{\leq, =\}$ in propositional decisions; indeed, if A is numeric, the natural propositional decision is of the type $A \leq a$:

$$InfoSplit(A, a, \bowtie, \mathcal{D}) = \sum_{i=1}^{i=2} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} Info(\mathcal{D}_i), \tag{19}$$

where $\bowtie \in \{\leq, =\}$, and:

$$InfoAtt(A, \mathcal{D}) = \min_{\bowtie\in\{\leq,=\}, a\in dom(A)} \{InfoSplit(A, a, \bowtie, \mathcal{D})\}. \tag{20}$$

Incidentally, the obtained tree gives a new kind of information, that is, the values of the splitting points that give the most information. If we consider a temporal data set $\mathcal{T}$, in which multivariate time series are non-discretized, we obtain a similar result in the temporal case by simply allowing, as above, $\bowtie \in \{\leq, =, >\}$; we may call the resulting algorithm *Temporal C4.5*. Observe that, as explained in the previous section, Temporal C4.5 has two new parameters, that is:

$$InfoSplit(A, X, a, \bowtie, \alpha, z, \mathcal{T}) = \sum_{i=1}^{i=2} \frac{|\mathcal{T}_i|}{|\mathcal{D}|} Info(\mathcal{T}_i), \tag{21}$$

where X takes values in $\mathcal{X} \cup \{eq\}$, and T_1, T_2 are computed as explained in the previous section (but, this time, varying $\alpha \in (0, 1]$ and $z \geq 0$), and

$$InfoAtt(A, \mathcal{T}) = \min_{\substack{X\in\mathcal{X}\cup\{eq\},\alpha\in(0,1],\\ 0\leq z\leq Max_z, a\in dom(A)}} \{InfoSplit(A, X, a, \bowtie, \alpha, z, \mathcal{T})\}, \tag{22}$$

where Max_z must be fixed beforehand.

Table 1 A summary of resampled datasets from [2].

Dataset	Train cases	Test cases	Channels	Length	Classes
AtrialFibrillation (AF)	24	6	2	150	3
FingerMovements (FM)	104	26	28	50	2
Libras (LI)	180	45	2	45	15
LSST (LS)	168	42	6	36	14
NATOPS (NA)	96	24	24	51	6
RacketSports (RS)	96	24	6	30	4
SelfRegulationSCP1 (S1)	96	24	6	150	2
SelfRegulationSCP1 (S2)	96	24	7	150	2
UWaveGestureLibrary (UW)	96	24	3	150	8

A working implementation. The open-source learning suite Weka [15] offers the implementation the algorithm C4.5 in Java, called *J48*. The most important distinctive characteristic of one specific implementation over the others is the stopping condition; different stopping conditions may lead to different trees. J48 uses a very intuitive principle: having decided a minimal *purity degree* (i.e., a minimal value of $Info(\mathcal{D})$, where $\mathcal{D}$ is associated to a leaf), the base case of the learner is fired on a node if its purity degree is high enough.

Being object-oriented, such an implementation is ideal to test the predictive capabilities of the trees learned following the schema in Section 3 with minimal (yet, non-trivial) modifications. As a matter of fact, we can keep the entire structure of J48: model construction, stopping condition, and training/test performance indicator calculation and displaying. There are two main modifications required:

(i) input data representation, and

(ii) splitting management.

As far as the first point is concerned, we used an internal representation based on the string data type, that implicitly assumes that all temporal attributes have the same length and that there are no missing data. Strings have a simple internal structure, in which each value is separated from the next one by a semicolon. Splitting, on the other hand, is taken care by simply building the necessary Java classes that take care of the possible cases. *Temporal J48*, as we call the resulting implementation, requires the following parameters in addition to those already required by J48:

(i) the value of α (which in this first experiment we did not optimized at each decision, unlike the general theory suggests);

(ii) the value of Max_z;

(iii) the reference intervals policy;

(iv) the subset of the language HS that one allows during the learning phase.

5 Experiments and Results

Data sets. In order to design a first systematic test aimed to establish the predictive capabilities of Temporal J48, we considered the public temporal data set from [2]. From it, we have extracted nine data sets, which contain problems that vary from the medical context, to automatic recognition of sing language words, to classification of different racket sports based on the movements performed by the athletes. Some adaptations were necessary, taking into account two aspects:

Table 2 Test results in terms of accuracy. Underlined results are the best ones in the group, and starred results are the absolute best ones.

Dataset	AF	FM	LI	LS	NA	RS	S1	S2	UW
J48 $1,0,0,0$	$\underline{83.33}$	$\underline{50.00}$	40.00	30.95	$\underline{79.17}$	70.83	$\underline{66.67}$	50.00	$\underline{66.67}$
J48 $1,1,0,0$	$\underline{83.33}$	42.31	51.11	30.95	75.00	$\underline{87.50}^*$	$\underline{66.67}$	54.17	62.50
J48 $1,1,1,1$	$\underline{83.33}$	42.31	$\underline{64.44}$	$\underline{38.10}$	62.50	79.17	$\underline{66.67}$	$\underline{62.50}$	54.17
ED_I	83.33	$\underline{76.92}$	86.67	$\underline{42.86}^*$	70.83	79.17	66.67	$\underline{66.67}$	87.50
DTW_I	$\underline{100.00}^*$	65.38	$\underline{91.11}^*$	33.33	$\underline{87.50}^*$	75.00	66.67	$\underline{66.67}$	91.67
DTW_D	83.33	57.69	$\underline{91.11}^*$	40.48	$\underline{87.50}^*$	$\underline{83.33}$	$\underline{83.33}^*$	$\underline{66.67}$	$\underline{95.83}^*$
T. J48 0.5	66.67	57.69	$\underline{80.00}$	23.81	$\underline{83.33}$	70.83	$\underline{83.33}^*$	54.17	62.50
T. J48 0.6	66.67	57.69	71.11	$\underline{26.19}$	79.17	$\underline{79.17}$	66.67	$\underline{75.00}^*$	58.33
T. J48 0.7	66.67	53.85	73.33	23.81	75.00	66.67	66.67	66.67	62.50
T. J48 0.8	$\underline{83.33}$	$\underline{80.77}^*$	75.56	$\underline{26.19}$	75.00	62.50	66.67	62.50	$\underline{66.67}$
T. J48 0.9	66.67	$\underline{80.77}^*$	71.11	23.81	66.67	62.50	66.67	70.83	$\underline{66.67}$

(i) the intrinsic computational inefficiency of Temporal J48 compared with existing methods, due to the substantial amount of information that can be extracted from its results, and
(ii) the intrinsic unbalance between training and test cardinalities in the original settings in [2].

We modified the data sets as the result of several initial tests by:
(i) trimming the number of temporal points for those data sets with too long time series, and, in particular, by limiting all time series to $N = 150$, and
(ii) re-sampling training and test instances to obtain a more standard $80\% - 20\%$ ratio.

The resulting situation is summarized in Table 1.

Experimental settings, results, and discussion. We tested the effectiveness of Temporal J48 against feature-based and distance-based methods, in terms of test accuracy only. This is a highly limited comparison, as the major strength of our approach lies in the interpretability of the (temporal component of the) resulting model, and such a characteristics does not emerge from a purely numeric performance test. Yet, it is interesting to see how good Temporal J48 performs against non-interpretable methods. The results are summarized in Table 2: underlined results are the best ones for the category (feature-based, distance-based, or symbolic), and starred results are the absolute best ones. As for feature-based models, we considered the standard J48 executed on three combinations of abstractions of the temporal data set; for each channel or attribute we computed mean, standard deviation, skewness, and kurtosis, and we combined them in three different ways, each expressed in Table 2 as a bit mask. So, for example, J48 with mask 1,1,0,0 means running the standard decision tree extraction algorithm on a abstracted data set with exactly two attributes per channel, namely mean and standard deviation. As for distance-based methods, we considered the standard, open-source available methods ED_I, DTW_I, and DTW_D, which require no parametrization. Finally, the parameter that we have used for Temporal J48 are: $0.5 \leq \alpha \leq 0.9$, with 0.1 step, $Max_z = 0$, and full HS.

As it can be seen, different temporal data set are best dealt with different approaches. In five out nine cases distance-based methods behaved best; in one case feature-based behaved best, using, in particular, only mean and standard deviation. In all other cases, three, there was a run of Temporal J48 that performed better than every other method. As we have explained, obtaining the highest accuracy was *not* our initial motivation; nonetheless, comparing our method against the others in terms of (test) accuracy is a good proxy for its performance. Yet, as a matter of fact, our approach overcame our expectations: we obtained an interpretable classification model of each of the data sets, and in three cases our model was also the most performing one, indicating that our approach is worth pursuing further. In this first experiment we did not examined all possible parametrization that Temporal J48 offers; in particular, decisions were taken only on the 0-th derivative (so no trends, and no acceleration/decelerations of trends were taken into account), and the uncertainty value α was fixed at the same value for all intervals. This suggests that a further analysis of the predictive capabilities of Temporal J48 may result in even better performances. More importantly, in some of the cases, Temporal J48 obtained a nearly perfect classification of some of the classes (i.e., individual ROC curve close to 1); these cases give rise to temporal formulas (which can be read on the resulting tree) which, in a way, describe those classes from the temporal logic point of view.

6 Conclusions

In this paper we approached the problem of multivariate time series classification. Existing methods for classification of multivariate time series present good performances in terms of accuracy, but the extracted models are not interpretable, in particular in the temporal component. Distance-based methods are based on the concept of distance between series, and feature-based methods, while compatible with interpretable classifiers, flatten the temporal component of the data set. Based on a recently proposed algorithm (Temporal ID3), which is able to classify previously discretized multivariate time series, we developed Temporal C4.5 and realized its implementation, Temporal J48, following the same principle of describing time series using interval temporal logic. Temporal J48 is compatible with the well-known data mining suite Weka.

The initial results show that the interval temporal logic HS is able to correctly describe the behaviour of multivariate time series. As a matter of fact, the predictive capabilities of Temporal J48 are comparable with those of existing methods, and superior to them in some cases, notwithstanding the fact that temporal decision tree models are interpretable even in the temporal component, and therefore undergo a very constrained learning phase (unlike non-interpretable methods, which are notoriously more adaptable). This suggests that temporal symbolic learning may be a promising topic, taking into account that, by examining the temporal component in an explicit way, several new learning parameters emerge that can be adjusted to improve the performances of the extracted models. Future developments of this line include, but are not limited to, exploring the predictive capabilities of variants of the language HS, and studying the possibility of adapting other well-known symbolic learning algorithms in the same way.

References

1 J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

2 A. Bagnall, H.A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. The UEA multivariate time series classification archive, 2018. `arXiv:1811.00075`.

3 A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

4 A. Brunello, E. Marzano, A. Montanari, and G. Sciavicco. J48SS: A novel decision tree approach for the handling of sequential and time series data. *Computers*, 8(1):21, 2019.

5 A. Brunello, G. Sciavicco, and I.E. Stan. Interval temporal logic decision tree learning. In *Proc. of the 16th European Conference on Logics in Artificial Intelligence (JELIA)*, volume 11468 of *Lecture Notes in Computer Science*, pages 778–793. Springer, 2019.

6 T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

7 B.D. Fulcher and N.S. Jones. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):3026–3037, 2014.

8 J.Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.

9 L. Hyafil and R.L. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15–17, 1976.

10 J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

11 J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

12 J.R. Quinlan. Simplifying decision trees. *International Journal of Human-Computer Studies*, 51(2):497–510, 1999.

13 G. Sciavicco, I.E. Stan, and A. Vaccari. Towards a general method for logical rule extraction from time series. In *Proc. of the 8th International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC)*, volume 11487 of *Lecture Notes in Computer Science*, pages 3–12. Springer, 2019.

14 M. Shokoohi-Yekta, J. Wang, and E. Keogh. On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In *Proc. of the 15th SIAM International Conference on Data Mining*, pages 289–297, 2015.

15 I.H. Witten, E. Frank, and M.A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 4th edition, 2017.

16 Junfeng Wu, Li Hua Yao, and Bin Liu. An overview on feature-based classification algorithms for multivariate time series. In *Proc. of the 3rd IEEE International Conference on Cloud Computing and Big Data Analysis*, pages 32–38, 2018.

A Appendix

Listing 1 One of the Temporal J48 models trained on data set RacketSports.

```
<L> var5 <= −2.756591
|   <InvA> var5 <= 0.308951
|   |   <=> var2 > −0.916901
|   |   |   <InvB> var0 <= 2.832243: Badminton_Clear (6.0)
|   |   |   [InvB] var0 > 2.832243: Badminton_Smash (1.0)
|   |   [=] var2 <= −0.916901
|   |   |   <B> var3 <= −0.207743
|   |   |   |   <InvB> var0 > 4.115426
|   |   |   |   |   <D> var0 > 1.452113: Squash_ForehandBoast (3.0)
|   |   |   |   |   [D] var0 <= 1.452113: Squash_BackhandBoast (1.0)
|   |   |   |   [InvB] var0 <= 4.115426
|   |   |   |   |   <InvB> var0 <= −0.215688: Badminton_Smash (2.0)
|   |   |   |   |   [InvB] var0 > −0.215688: Badminton_Clear (3.0)
|   |   |   [B] var3 > −0.207743: Squash_ForehandBoast (14.0)
|   [InvA] var5 > 0.308951
|   |   <InvB> var5 <= −2.27452
|   |   |   <InvA> var0 <= −1.044682: Squash_BackhandBoast (3.0/1.0)
|   |   |   [InvA] var0 > −1.044682: Squash_ForehandBoast (7.0)
|   |   [InvB] var5 > −2.27452: Squash_BackhandBoast (21.0)
[L] var5 > −2.756591
|   <A> var0 <= 0.098773
|   |   <InvB> var0 > −0.960139
|   |   |   <B> var4 <= 0.625893: Badminton_Smash (16.0)
|   |   |   [B] var4 > 0.625893: Badminton_Clear (1.0)
|   |   [InvB] var0 <= −0.960139: Badminton_Clear (2.0)
|   [A] var0 > 0.098773
|   |   <L> var4 > 8.703901: Badminton_Smash (4.0)
|   |   [L] var4 <= 8.703901: Badminton_Clear (12.0)
```

Time series classification methods. We briefly review the literature of time series classification. We used some of the available techniques, described here, for a comparison against Temporal J48.

Univariate time series classification is a well-studied problem in the literature; the reader can refer to [3] for an in-depth analysis on state-of-the-art methods for univariate time series classification. For the multivariate case, the classical accepted methods in the literature are *feature-based* or *distance-based.* Feature-based methods are very simple to understand as they are based on extracting numerical or categorical descriptions from each channel. Such descriptions can be simple statistical values (e.g., mean, minimum, maximum, variance, skewness) or yes/no values that refer to the presence of certain patterns (e.g., shapelets). The collection of all descriptions can be then used as input to a classical, static learning algorithm [16]. In some cases, algorithms can be adapted to natively extract patterns from the temporal data sets, as it is the case of [4]. The most widely accepted distance-based methods for multivariate time series classification is the classical *Nearest Neighbour* (NN) algorithm [6] equipped with a proper notion of distance. In the univariate case, given two time series $T_1 = a_1, a_2, \ldots, a_N$ and $T_2 = b_1, b_2, \ldots, b_N$, the *Euclidean distance (ED)* between T_1 and T_2 is simply the sum of the Euclidean distance between each pair (a_i, b_i). The *dynamic time warping distance (DTW)* generalizes such concept by means of an alignment procedure that consists in constructing a $N \times N$ *distance matrix*, computed via dynamic programming, that allows one to find the alignment that minimizes the point-to-point Euclidian distance. In other words, DTW generalizes the notion of Euclidean distance from single points to single time series. In the multivariate case, in [14] this notion of distance is further generalized in two versions, named DTW_I (*independent* DTW) and DTW_D (*dependent* DTW), which differ by how the different channels are combined into a single distance. Thus, distance-based multivariate time series classification is traditionally solved via ED$_I$ (the independent multivariate generalization of the Euclidean distance), DTW$_I$, or DTW$_D$. Feature-based and distance-based methods present similar drawbacks: feature-based extract an explicit theory of a temporal data set, but such a theory is hardly interpretable, as it is written in the language of abstract indicators, and non-temporal, as the temporal component plays no role, while distance-based methods solve the classification problem in a black-box way, without extracting any symbolic theory at all. So, the methods of both groups are, in a way, non-interpretable. Finally, using Temporal ID3 (Section 2) to classify previously discretized multivariate time series can be thought of as a *timeline-based* classification method. In this particular taxonomy, Temporal C4.5 is a *symbolic* time series classification method.

Temporal J48 internal representation. In Figure 3, we see how data are represented in Temporal J48. Data are abstractly represented as at the top of the figure; the same data present naturally as a matrix, as at the bottom-left of the figure. Finally, we internally represent using a *string* data type, as in the bottom-right of the figure.

A decision tree learned by Temporal J48. Let analyze, more in depth, some of the results of our test with Temporal J48.

Consider the temporal data set RacketSports (see Section 5), in which each multivariate time series describes the movements of an athlete playing badminton or squash whilst wearing a smart watch, which relayed the X, Y, Z coordinates for both a gyroscope and an accelerometer to a smart phone. More in particular, four classes are identified, two movements during the activity of playing squash, that is, *back-hand boast* and *fore-hand*

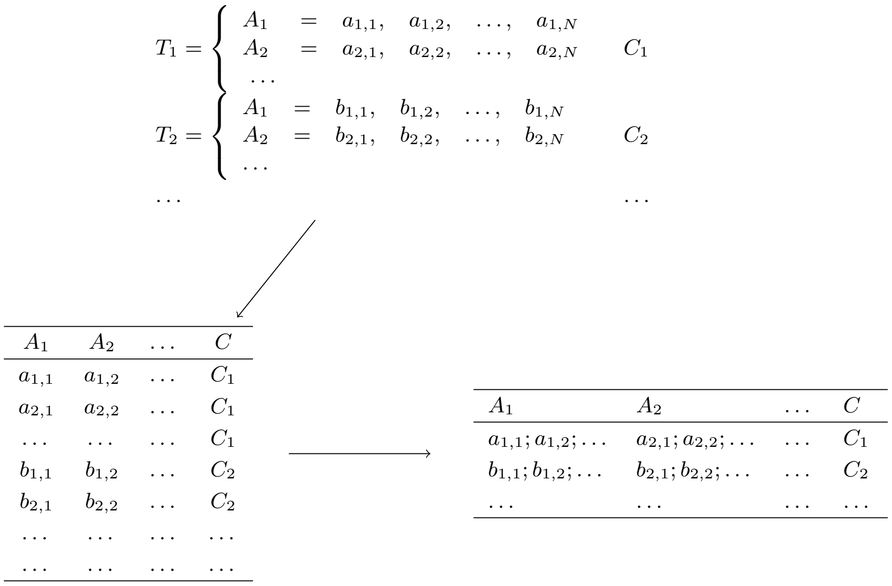
$T_1 = \begin{cases} A_1 = a_{1,1}, a_{1,2}, \ldots, a_{1,N} \\ A_2 = a_{2,1}, a_{2,2}, \ldots, a_{2,N} \\ \ldots \end{cases}$ C_1
$T_2 = \begin{cases} A_1 = b_{1,1}, b_{1,2}, \ldots, b_{1,N} \\ A_2 = b_{2,1}, b_{2,2}, \ldots, b_{2,N} \\ \ldots \end{cases}$ C_2
$\ldots$ $\ldots$
A_1 A_2 $\ldots$ C
$a_{1,1}$ $a_{1,2}$ $\ldots$ C_1
$a_{2,1}$ $a_{2,2}$ $\ldots$ C_1
$\ldots$ $\ldots$ $\ldots$ C_1
$b_{1,1}$ $b_{1,2}$ $\ldots$ C_2
$b_{2,1}$ $b_{2,2}$ $\ldots$ C_2
$\ldots$ $\ldots$ $\ldots$ $\ldots$
$\ldots$ $\ldots$ $\ldots$ $\ldots$
A_1 A_2 $\ldots$ C
$a_{1,1}; a_{1,2}; \ldots$ $a_{2,1}; a_{2,2}; \ldots$ $\ldots$ C_1
$b_{1,1}; b_{1,2}; \ldots$ $b_{2,1}; b_{2,2}; \ldots$ $\ldots$ C_2
$\ldots$ $\ldots$ $\ldots$ $\ldots$

Figure 3 Representation of a temporal data set: original data set (top), natural, tabular representation (bottom, left), and Temporal J48 internal representation (bottom, right).

Table 3 Test performances for the RacketSports data set.

TP	FP	Prec.	Rec.	F-M	MCC	ROC	PRC	Class
0.66	0.05	0.80	0.66	0.72	0.65	0.80	0.61	Smash, badminton
0.83	0.11	0.71	0.83	0.76	0.68	0.86	0.63	Clear, badminton
0.66	0.00	1.00	0.66	0.80	0.77	0.83	0.75	Fore-hand boast, squash
1.00	0.11	0.75	1.00	0.85	0.81	0.94	0.75	Back-hand boast, squash

boast, and two movement during the activity of playing badminton, that is, *smash* and *clear*. These movements are described by six channels, which contain the values of the sensors attached for each physical dimension at each moment of time. These variables are codified as follows: Var_0, Var_1, and Var_2 (resp., Var_3, Var_4, and Var_5) are the gyroscope (resp., accelerometer) values for X, Y and Z. Run with $\alpha = 0.6$ on this data set, Temporal J48 returned the tree in Listing 1, which has, in test phase, the performances shown in Tab. 3. By focusing on the squash back-hand boast movement only, which has a 0.94 ROC area, one can extract from Listing 1 a formula of HS, shown in Figure 4, that describes such a movement along the temporal component. This proves that our model extraction method allows one to *interpret* the underlying theory. In opposition, feature-based methods flatten the temporal component, so while a symbolic theory is extracted, it is not temporal, and distance-based methods do not extract a theory at all.

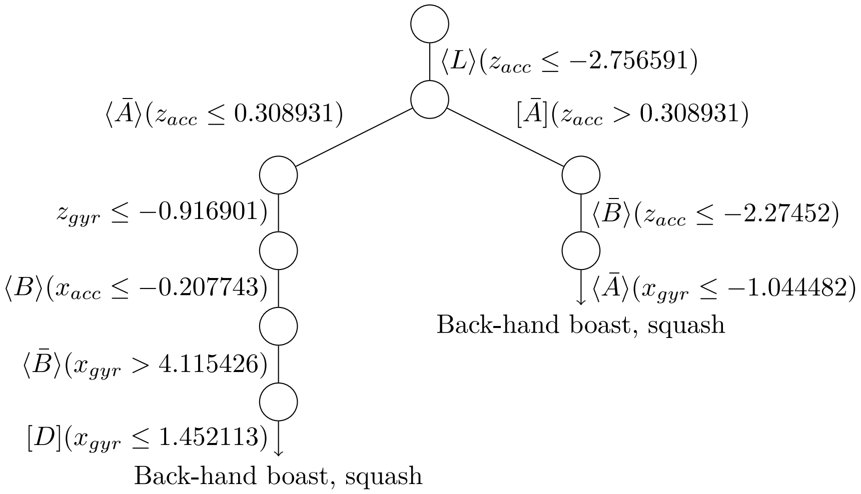
$\langle L\rangle(z_{acc} \leq -2.756591)$
$\langle \bar{A}\rangle(z_{acc} \leq 0.308931)$
$[\bar{A}](z_{acc} > 0.308931)$
$z_{gyr} \leq -0.916901)$
$\langle \bar{B}\rangle(z_{acc} \leq -2.27452)$
$\langle B\rangle(x_{acc} \leq -0.207743)$
$\langle \bar{A}\rangle(x_{gyr} \leq -1.044482)$
Back-hand boast, squash
$\langle \bar{B}\rangle(x_{gyr} > 4.115426)$
$[D](x_{gyr} \leq 1.452113)$
Back-hand boast, squash

Figure 4 Extracted theory for the movement of back-hand boast during squash.