Interval Temporal Random Forests with an Application to COVID-19 Diagnosis

Federico Manzella
Dept. of Mathematics and Computer Science, University of Ferrara, Italy

Giovanni Pagliarini
Dept. of Mathematics and Computer Science, University of Ferrara, Italy
Dept. of Mathematical, Physical, and Computer Sciences, University of Parma, Italy

Guido Sciavicco
Dept. of Mathematics and Computer Science, University of Ferrara, Italy

Ionel Eduard Stan
Dept. of Mathematics and Computer Science, University of Ferrara, Italy
Dept. of Mathematical, Physical, and Computer Sciences, University of Parma, Italy

Abstract

Symbolic learning is the logic-based approach to machine learning. The mission of symbolic learning is to provide algorithms and methodologies to extract logical information from data and express it in an interpretable way. In the context of temporal data, interval temporal logic has been recently proposed as a suitable tool for symbolic learning, specifically via the design of an interval temporal logic decision tree extraction algorithm. Building on it, we study here its natural generalization to *interval temporal random forests*, mimicking the corresponding schema at the propositional level. Interval temporal random forests turn out to be a very performing multivariate time series classification method, which, despite the introduction of a functional component, are still logically interpretable to some extent. We apply this method to the problem of diagnosing COVID-19 based on the time series that emerge from cough and breath recording of positive versus negative subjects. Our experiment show that our models achieve very high accuracies and sensitivities, often superior to those achieved by classical methods on the same data. Although other recent approaches to the same problem (based on different and more numerous data) show even better statistical results, our solution is the first logic-based, interpretable, and explainable one.

2012 ACM Subject Classification Computing methodologies → Machine learning algorithms

Keywords and phrases Interval temporal logic, decision trees, random forests, sound-based diagnosis

Digital Object Identifier 10.4230/LIPIcs.TIME.2021.7

Acknowledgements We thank the INdAM GNCS 2020 project *Strategic Reasoning and Automated Synthesis of Multi-Agent Systems* for partial support, the PRID project *Efforts in the uNderstanding of Complex interActing SystEms*, the University of Udine (Italy), the University of Gothenburg (Sweden), and the Chalmers University of Technology (Sweden) for providing the computational resources, and the University of Cambridge (UK) for sharing their data. Moreover, the open access publication of this article was supported by the Alpen-Adria-Universität Klagenfurt, Austria.

1 Introduction

Machine Learning (ML) is at the core of modern Artificial Intelligence. It can be defined as the process of automatically extracting the theory that underlies a phenomenon, and expressing it in machine-friendly terms, so that it can be later used in applications. The potential of ML is limitless, and it ranges from learning rules that classify patients at some risk, to formalizing the factors that influence pollution in a certain area, to recognizing voices, signatures, images, and many others. The most iconic and fundamental separation between the sub-fields of ML is the one between *functional* and *symbolic* learning. Functional

28th International Symposium on Temporal Representation and Reasoning (TIME 2021).
Editors: Carlo Combi, Johann Eder, and Mark Reynolds; Article No. 7; pp. 7:1–7:18
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

learning is the process of learning a *function* that represents such underlying theory; functions can be as simple as *linear functions*, or as complex as *deep neural networks*. Symbolic learning, on the other hand, is the process of learning a *logical description* that represents a phenomenon. Symbolic learning is sometimes statistically less accurate than functional one, but its results can be interpreted and explained by humans, while, in general, functional models are considered *black-boxes*. Until very recently, symbolic learning models were limited by their underlying logical language, that is, propositional logic, and temporal, spatial, and, in general, non-propositional data were usually dealt with propositional learning by *flattening* the non-propositional dimension using global features (e.g., the average temperature instead of all values within the monitored period). This resulted in the possibility of using off-the-shelf methods after a phase of data abstraction, but severely hampered the interpretability of the results, and in many cases the statistical performances of the extracted model as well.

Interval temporal logic decision trees are a first step in the direction of improving the expressive power of symbolic methods by replacing propositional logic with a more expressive formalism in a classical, well-known schema. They were introduced in [13, 38], and have shown great potential as a method to classify multivariate time series. Temporal logic decision trees are part of a bigger project that aims to establish a whole new area, generally known as *modal symbolic learning*, whose driving principle is precisely the study of symbolic learning schemata and methods based on propositional modal logic. Propositional decision trees can be generalized into *bags of trees* and then into *random forests* [7] to obtain classifiers based on several trees instead of a single one. Sets of trees tend to be more performing than single trees, and while they are considered at the verge between symbolic and functional learning, their symbolic nature is still evident: sets of trees, as single trees, can be analyzed and discussed, and, although the process of extracting rules is not as immediate as in single trees, is still possible [17, 31]. Building on this idea, in this paper we present an approach to *interval temporal logic random forests*, based, as single temporal logic trees are, on interval temporal logic. Interval temporal forests follow the same principles as the propositional ones: as a general rule, a forest is a schema based on the idea that different trees are built from different subsets of the training set and different subsets of the attributes; in the temporal case, moreover, they may differ by the subset of interval relations that are allowed in the learning (in [29], the problem of selecting subsets of relations in the learning phase, treated as feature selection problem, has been studied). We use interval temporal random forests in the same way as interval temporal decision trees, that is, to solve multivariate time series classification problems. Classification of time series is an active area of research: air quality control and prediction in climate science, prices and rates of inflation in economics, infectious diseases trends and spreading patterns in medicine, pronunciation of word signs in linguistics, sensor recordings of systems in aerospace engineering, among many others, are all problems that can be expressed in terms of time series classification. But the intrinsic versatility of time series allows us to act on less immediate applications, including, for example, interpreting *breath* and *cough* recordings as a medium for diagnosis of respiratory diseases. *COVID-19* is a respiratory disease onset by the virus *SARS-CoV2*, classified in 2019, which has caused a pandemic in the years 2020 and 2021; the current literature on this topic is huge, and spans every field from the medical, economical, sociological, up to and including applications of artificial intelligence. In early 2020, via an application and a website specifically built to this purpose, several cough and breath sound samples were recorded from anonymous volunteers, along with a small medical history and their declaration on positivity/negativitiy w.r.t. a recent COVID-19 test. Such recordings were used in [11] to train a classical, functional, adimensional classification system with the purpose of designing a prototype automatic diagnosis system. The data were also made publicly available. In this paper, we use the same recordings by treating them as multivariate time series, and we test the interval temporal logic random forest model on them, serving two purposes: contributing to the fight against the pandemic with yet another system for automatic diagnosis and proving the effectiveness of our model.

This paper is organized as follows. In Section 2 we give some necessary background on time series and their classification, interval temporal logic, and sound-based diagnosis of respiratory diseases. In Section 3 we briefly recall interval temporal decision trees and introduce interval temporal random forests. Then, in Section 4 we discuss the data, the applied transformations, the experimental setup, and the results, before concluding.

2 Background

Time series and their classification. A *time series* T is a set of $n \geq 1$ variables that evolve over time, where each variable is an ordered collection of N_T numerical or categorical values described as follows:

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

A time series is called *multivariate*, if $n > 1$; otherwise it is *univariate*. A univariate time series with categorical values is also known as a *time* (or *temporal*) *sequence*; we use the term *time series* to denote multivariate, mixed (numerical and categorical) set of temporal variables. Categorical values are fairly uncommon in time series, and typical temporal data sets are usually numerical. A *temporal data set* is a set $\mathcal{T} = \{T_1, \ldots, T_m\}$ of m temporal *instances* defined over a set of n attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$, each of which is a univariate time series T having N_T points. A *categorical labeled* temporal data set is a temporal data set where the instances are associated to a *target variable* $\mathcal{C} = \{C_1, \ldots, C_l\}$, also known as *class variable*. In this paper, we assume that temporal data sets have no missing values, or that missing values are simply substituted by placeholders. Implicitly, we are also assuming that temporal attributes are sampled at the same granularity.

Time series are very versatile and we can use them to represent very diverse situations. For example, we may want to describe the clinical history of a patient; the set of numerical attributes may include *fever* and *level of pain*, which change along time on a scale, say, of minutes, and the problem may be to distinguish between *relapsing* and *non-relapsing* patients. Similarly, we can use time series to describe the behaviour of complex machines to which sensors are attached; they may measure *temperature* and *pressure* on a scale of seconds, and the problem may be to distinguish between machines that *need* from those which *do not need* preemptive maintenance. The *multivariate time series classification* is the problem of finding a formula or a set of formulas (i.e., *symbolic classification*), or a function (i.e., *functional classification*) that associates multivariate time series to classes. Several approaches for time series classification have been proposed in the literature, that span from purely functional ones [16, 24, 25, 30, 39], to distance-based ones [28], to symbolic ones [2, 3, 15, 41], to shapeled-based ones [12].

Interval temporal logic. While several different interval temporal logics have been proposed in the recent literature [18], *Halpern and Shoham's (HS)* [19] is certainly the formalism that has received the most attention, being the most natural modal logic for temporal intervals. From a logical point of view, HS and its fragments have been studied on the most important classes of linearly ordered sets, from the class of all linear orders, to the classes of linear orders that can be built on classical sets such as $\mathbb{N}, \mathbb{Q}$ and $\mathbb{R}$ [9, 10, 19]. Nevertheless, from the learning point of view, temporal (and static) data sets are finite, fully represented structures; therefore, we focus our attention on finite domains. Let $[N]$ be a finite, initial subset of $\mathbb{N}_+$ of cardinality $N > 1$, that is, $[N] = \{1, 2, \ldots, N\}$. A *strict interval* over $[N]$ is an ordered pair $[x, y]$, where $x, y \in [N]$ and $x < y$. If we exclude the identity relation, there are 12 different binary ordering relations between two strict intervals on a linear order, often called *Allen's interval relations* [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 Tab. 1, and their *inverses*, that is, $R_{\overline{X}} = (R_X)^{-1}$, for each $X \in \{A, L, B, E, D, O\}$. We interpret interval structures as Kripke structures, with Allen's relations playing the role of accessibility relations. Thus, we associate an *existential modality* $\langle X \rangle$ with each Allen's relation R_X. Moreover, for each $X \in \{A, L, B, E, D, O\}$, the *transpose* of modality $\langle X \rangle$ is modality $\langle \overline{X} \rangle$ corresponding to the inverse relation $R_{\overline{X}}$ of R_X. Now, let $\mathcal{X} = \{A, \overline{A}, L, \overline{L}, B, \overline{B}, E, \overline{E}, D, \overline{D}, O, \overline{O}\}$; Halpern and Shoham's interval temporal logic (HS) [19] is a multi-modal logic with formulas built from a finite, non-empty set $\mathcal{AP}$ of *atomic propositions* (also referred to as *proposition letters*), the propositional connectives $\vee$ and $\neg$, and a modality for each Allen's interval relation, and well-formed formulas of HS are generated by the grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi,$$

where $p \in \mathcal{AP}$ and $X \in \mathcal{X}$. The other propositional connectives and constants (e.g., $\psi_1 \wedge \psi_2 \equiv \neg\psi_1 \vee \neg\psi_2, \psi_1 \rightarrow \psi_2 \equiv \neg\psi_1 \vee \psi_2$ and $\top = p \vee \neg p$), as well as, for each $X \in \mathcal{X}$, the *universal modality* $[X]$ (e.g., $[A]\varphi \equiv \neg\langle A \rangle \neg\varphi$), can be derived in the standard way.

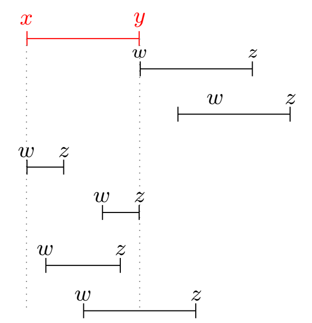
The strict semantics of HS is given in terms of *timelines* (or, more commonly, *interval models*) $T = \langle \mathbb{I}([N_T]), V \rangle$[1], where $[N_T] = \{1, 2, \ldots, N_T\}$ is a finite linear order, $\mathbb{I}([N_T])$ is the set of all *(strict) intervals* over $[N_T]$ with cardinality $N_T(N_T - 1)/2$, and V is a *valuation function* $V : \mathcal{AP} \rightarrow 2^{\mathbb{I}([N_T])}$ which assigns to every 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, denoted by $T, [x, y] \Vdash \varphi$, is defined by structural induction on the complexity of formulas as follows:

$T, [x, y] \Vdash p$	if and only if	$[x, y] \in V(p)$, for each $p \in \mathcal{AP}$;
$T, [x, y] \Vdash \neg\psi$	if and only if	$T, [x, y] \not\Vdash \psi$ (i.e., it is not the case that $T, [x, y] \Vdash \psi$);
$T, [x, y] \Vdash \psi_1 \vee \psi_2$	if and only if	$T, [x, y] \Vdash \psi_1$ or $T, [x, y] \Vdash \psi_2$;
$T, [x, y] \Vdash \langle X \rangle \psi$	if and only if	there exists $[w, z]$ s.t. $[x, y] R_X [w, z]$ and $T, [w, z] \Vdash \psi$;

where $X \in \mathcal{X}$. Given a model $T = \langle \mathbb{I}([N_T]), V \rangle$ and a formula φ, we say that T *satisfies* φ if there exists an interval $[x, y] \in \mathbb{I}([N_T])$ such that $T, [x, y] \Vdash \varphi$. A formula φ is *satisfiable* if there exists an interval model that satisfies it. Moreover, a formula φ is *valid* if it is satisfiable on every interval model or, equivalently, if its negation $\neg\varphi$ is *unsatisfiable*.

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

Table 1 Allen's interval relations and HS modalities.

HS modality	Definition w.r.t. the interval structure	Example
$\langle A \rangle$ (after)	$[x, y]R_A[w, z] \Leftrightarrow y = w$	
$\langle L \rangle$ (later)	$[x, y]R_L[w, z] \Leftrightarrow y < w$	
$\langle B \rangle$ (begins)	$[x, y]R_B[w, z] \Leftrightarrow x = w \wedge z < y$	
$\langle E \rangle$ (ends)	$[x, y]R_E[w, z] \Leftrightarrow y = z \wedge x < w$	
$\langle D \rangle$ (during)	$[x, y]R_D[w, z] \Leftrightarrow x < w \wedge z < y$	
$\langle O \rangle$ (overlaps)	$[x, y]R_O[w, z] \Leftrightarrow x < w < y < z$	

Sounds and respiratory diseases. Human sounds, both audible (such as cough and breath) and non-audible (such as heartbeat) have been long considered useful tools for diagnosis. The recent use of sounds, and, in particular, cough, for the automatic, ML-based, diagnosis of respiratory diseases includes many examples (see [35] for a systematic review). Because of the current sanitary emergency, the case of COVID-19 has received a lot of attention. Very recent approaches include [20, 23, 26] and [11], from which the data used in this paper are borrowed. A common denominator to these approaches is their classical treatment of the dimensional data, such as breath and cough, that consists of extracting (even complex) numerical attributes from them, and then using such attributes to train classical classifier(s). Moreover, following the recent trends, the existing systems share the use of functional, complex classification learning systems (mainly deep neural networks). As it often happens, such black-box solutions tend to outperform symbolic approaches in terms of accuracy, but they give up, at the same time, the possibility of interpreting, discussing, and explaining the models. We, on the contrary, treat cough and breath as time series, and use native temporal methods to classify them. This results in an interesting compromise: our numerical performances are comparable to those of functional classifiers, and yet our models are explicit. While extracting a *theory of the sound of COVID-19* is outside the scope of this paper, the fact remains that we are able to pinpoint interval temporal logic formulas that in some way describe the differences between negative and positive cases; these formula may be even translated into diagnostic principles, and, in this particular case, audible sounds.

3 A Theory of Temporal Decision Trees and Forests

Temporal decision trees. Let $\mathcal{T}$ be a temporal data set described by n attributes $\{A_1, \ldots, A_n\}$. Given a time series $T \in \mathcal{T}$ and a time point t, we denote by $A(t)$ the value of A at the point t, and by $dom(A)$ the domain of A. A temporal data set entails a propositional alphabet $\mathcal{AP}$ defined as follows:

$$\mathcal{AP} = \{A \bowtie_{\sim\gamma} a \mid A \in \mathcal{A}, \bowtie \in \{<, \leq, =, \geq, >\}, \sim \in \{<, \leq, \geq, >\} \text{ and } a \in dom(A)\}.$$

The set $\mathcal{AP}$ is the natural generalization of the set of propositional letters that implicitly emerges in inductive processes from static data (e.g., *fever greater than* 38 *degrees*). The main difference between the two cases, propositional and temporal, is that in the latter case propositions in $\mathcal{AP}$ are given an interval semantics, that is, they are evaluated over intervals of time; this is a natural choice that depends from the fact that time series describe continuous processes, in which evaluations based on punctual values have little sense. Intuitively, consider an interval of time $[x, y]$ and an attribute A that varies on it. We can ask the question $A \bowtie a$ over the entire interval, which is positively answered if *every value* of A in the interval $[x, y]$ respects the given constraint; but, to enhance an interval-based semantics we soften the meaning of $A \bowtie a$ using the expression $A \bowtie_{\sim\gamma} a$, with $\gamma \in (0, 1]$, that is interpreted as true if, assuming for example $\sim$ to be $\geq$, *at least the* γ *fraction* of the values in $[x, y]$ respect the given constraint. More formally, we say that:

$$T, [x, y] \Vdash A \bowtie_{\sim\gamma} a \quad \text{if and only if} \quad \frac{|\{z | x \leq z \leq y \text{ and } A(z) \bowtie a\}|}{y - x + 1} \sim \gamma.$$

In the particular case of propositional learning the set of propositional letter is complete, that is, for every letter p there exists a letter q that behaves as $\neg p$. The set $\mathcal{AP}$, as defined above, is complete as well in this sense; however, completeness is not necessary in the learning phase, and subsets of $\mathcal{AP}$ may be used to improve the experimental efficiency of the process. In the context of temporal decision trees and forests we do not ask if $A \bowtie_{\sim\gamma} a$ holds only in the *current interval* but also if *there exists an interval*, related to the current one, in which that holds. Thus, the language of temporal decision trees encompasses a set of *temporal existential decisions*:

$$\mathcal{S}_\Diamond = \{\langle X \rangle (A \bowtie_{\sim\gamma} a) \mid X \in \mathcal{X}, A \in \mathcal{A} \text{ and } a \in dom(A)\},$$

a set of *temporal universal decisions*:

$$\mathcal{S}_\Box = \{[X](A \bowtie_{\sim\gamma} a) \mid X \in \mathcal{X}, A \in \mathcal{A} \text{ and } a \in dom(A)\},$$

and a set of *atemporal decisions*:

$$\mathcal{S}_= = \{A \bowtie_{\sim\gamma} a \mid A \in \mathcal{A} \text{ and } a \in dom(A)\}.$$

Together, they form a set of *temporal and atemporal decisions* $\mathcal{S}$, defined as:

$$\mathcal{S} = \mathcal{S}_\Diamond \cup \mathcal{S}_\Box \cup \mathcal{S}_=.$$

So, binary *temporal decision trees* τ are formulas of the following grammar:

$$\tau ::= (S_= \wedge \tau) \vee (\neg S_= \wedge \tau) \mid (S_\Diamond \wedge \tau) \vee (\neg S_\Diamond \wedge \tau) \mid C,$$

where $S_= \in \mathcal{S}_=$ is an atemporal decision, $S_\Diamond \in \mathcal{S}_\Diamond$ is a temporal existential decision, and $C \in \mathcal{C}$ is a class. Thus, a temporal decision tree is a rooted tree whose leaves are labeled with classes, and whose edges are labeled with temporal or atemporal decisions. We denote by $root(\tau)$ the *root* of τ, and we use $\ell_1, \ell_2, \ldots$ (resp., $\nu_1, \nu_2, \ldots$) to denote the *leaves* (resp., *nodes*, both leaf and non-leaf ones). Each non-leaf node ν of τ has a *left* (resp., *right*) *child* $L(\nu)$ (resp., $R(\nu)$) whose edge is decorated with $S \in \mathcal{S}_= \cup \mathcal{S}_\Diamond$ (resp., $\neg S \in \mathcal{S}_= \cup \mathcal{S}_\Box$), each non-root node ν has a *parent* $P(\nu)$, and each leaf ℓ is labeled with a class, denoted by $C(\ell)$. A *path* of *length* h between two nodes of τ is a finite sequence of nodes $\nu_h, \nu_{h-1}, \ldots, \nu_0$ such that $\nu_{i+1} = P(\nu_i)$, for each $i = 0, \ldots, h-1$; if ν_h is the root $root(\tau)$ and ν_0 is a leaf ℓ, then the path $root(\tau) \rightsquigarrow \ell$ is called *branch*. In general, a path of length h is *decorated* with h temporal and atemporal decisions on its edges, denoted by $\nu_h \overset{S_h}{\rightsquigarrow} \nu_{h-1} \overset{S_{h-1}}{\rightsquigarrow} \ldots \overset{S_1}{\rightsquigarrow} \nu_0$, where $S_i \in \mathcal{S}$, for each $i = 1, \ldots, h$.

In order to define the semantics of temporal decision trees, we need the notions of temporal path-formula, satisfiability of a temporal path-formula, and temporal data set splitting. A *temporal path-formula* $\varphi_{\nu_h \rightsquigarrow \nu_0}$ of a path $\nu_h \overset{S_h}{\rightsquigarrow} \nu_{h-1} \overset{S_{h-1}}{\rightsquigarrow} \ldots \overset{S_1}{\rightsquigarrow} \nu_0$, where $S_i \in \mathcal{S}$, in a temporal decision tree τ, is inductively defined on h:

- if $h = 0$, then $\varphi_{\nu_0 \rightsquigarrow \nu_0} = \top$;
- if $h > 0$, then let $\varphi_{\nu_{h-1} \rightsquigarrow \nu_0} = \xi'_{h-1} \wedge \ldots \wedge \xi'_1 \wedge \top$, and let us call ξ'_i *positive* if it has the form $\xi'_i = \langle X \rangle (A \bowtie_\gamma a \wedge \psi_i), \xi'_i = (\langle X \rangle (A \bowtie_\gamma a) \wedge \psi_i), \xi'_i = (A \bowtie_\gamma a \wedge \psi_i)$ or $\xi' = (A \bowtie_\gamma a \wedge \psi_i)$, with $X \in \mathcal{X}$, and *negative* otherwise. Then $\varphi_{\nu_h \rightsquigarrow \nu_0}$ is defined by cases:
  - if $\nu_{h-1} = left(\nu_h)$, then $\varphi_{\nu_h \rightsquigarrow \nu_0} = S_h \wedge \xi_{h-1} \wedge \ldots \wedge \xi_1 \wedge \top$, where, for $1 \leq i \leq h-1$:
    - $\xi_i = \langle X \rangle (A \bowtie_\gamma a \wedge \xi'_i)$, if $S_h = \langle X \rangle (A \bowtie_\gamma a)$ and ξ'_i is positive;
    - $\xi_i = (A \bowtie_\gamma a \wedge \xi'_i)$, if $S_h = A \bowtie_\gamma a$ and ξ'_i is positive;
    - $\xi_i = (\langle X \rangle (A \bowtie_\gamma a) \wedge [X](A \bowtie_\gamma a \rightarrow \xi'_i))$, if $S_h = \langle X \rangle (A \bowtie_\gamma a)$ and ξ'_i is negative;
    - $\xi_i = (A \bowtie_\gamma a \rightarrow \xi'_i)$, if $S_h = A \bowtie_\gamma a$ and ξ'_i is negative;
  - if $\nu_{h-1} = right(\nu_h)$, then $\varphi_{\nu_h \rightsquigarrow \nu_0} = (S_h) \wedge \xi_{h-1} \wedge \ldots \wedge \xi_1 \wedge \top$, where, for $1 \leq i \leq h-1$,
    - $\xi_i = \xi'_i$, if ξ'_i is positive;
    - $\xi_i = (S_h \wedge \xi'_i)$, if ξ'_i is negative.

Temporal path-formulas generalize their propositional counterpart, where propositional path-formulas are simply conjunctions of the decisions. Now, we need to define how they are actually interpreted. In the static case, from a data set $\mathcal{D}^\nu$ associated to a node ν of a (static) decision tree τ one computes immediately the two data sets $\mathcal{D}^{L(\nu)}$ and $\mathcal{D}^{R(\nu)}$ that are entailed by a propositional decision $S \in \mathcal{S}$. In the temporal case, however, this step requires a bigger effort. We start by assuming that each temporal instance T is *anchored* to a set of intervals in the set $\mathbb{I}([N_T]) \cup [0,1]$, denoted $T.refs$. At the beginning of the learning phase, $T.refs = \{[0,1]\}$ for every T, where $[0,1]$ is an external interval that we add to the domain of every time series, and that we interpret as a privileged observation point from which the learning takes place. Temporal decision tree learning is a *local* learning process; the local nature of decision trees does not transpire at the static level, but it becomes evident at the modal one. Every decision entails, potentially, new reference intervals for every instance of a data set. In particular, given a time series T with associated $T.refs$, and given a decision S, we can compute a set of *new reference intervals* $f(T.refs, S)$ as:

$$\{[w,z] \in \mathbb{I}([N_T]) \mid \exists [x,y] \in T.refs \wedge [x,y] R_X [w,z] \wedge T,[w,z] \Vdash A \bowtie_{\sim\gamma} a\}$$

if $S = \langle X \rangle (A \bowtie_{\sim\gamma} a)$, and as:

$$\{[w,z] \in T.refs \mid T,[w,z] \Vdash A \bowtie_{\sim\gamma} a\}$$

if $S = A \bowtie_{\sim\gamma} a$. When S is clear from the context, we use $T.refs'$ to denote $f(T.refs, S)$. For a decision $S \in \mathcal{S}_= \cup \mathcal{S}_\Diamond$, we use the notation $T \Vdash S$ or $T, T.refs \Vdash S$ (respectively, $T \Vdash \neg S$ or $T, T.refs \Vdash \neg S$) to identify the members of $\mathcal{T}^{L(\nu)}$ (respectively, $\mathcal{T}^{R(\nu)}$). The notion of a time series satisfying a decision allows us to discuss the instance semantics of a temporal decision tree. Given a temporal decision tree τ and a temporal instance $T \in \mathcal{T}$ anchored to $T.refs$ at $root(\tau)$, the *class assigned by* τ *to* T, denoted by $\tau(T, T.refs)$, is inductively defined as:

$$\begin{array}{lll} C & \text{if} & \tau = C, \\ \tau^L(T, T.refs') & \text{if} & \tau = (S \wedge \tau^L) \vee (\neg S \wedge \tau^R) \text{ and } T, T.refs \Vdash S; \\ \tau^R(T, T.refs) & \text{if} & \tau = (S \wedge \tau^L) \vee (\neg S \wedge \tau^R) \text{ and } T, T.refs \Vdash \neg S; \end{array}$$

where $S \in \mathcal{S}_= \cup \mathcal{S}_\Diamond$. Moreover, we denote by $\tau(T) = \tau(T, \{[0,1]\})$, where $[0,1]$ is the privileged observation point; we call $\tau(T)$ the *instance semantics* of τ. As a whole, a temporal decision tree is interpreted over a labeled data set $\mathcal{T}$ via the *dataset semantic* relation $\Vdash_\theta$, which generalizes $\models$ from single instances to data sets. The parameter θ can represent any suitable measure of statistical performances of τ on $\mathcal{T}$, and it can be obtained by systematic application of the instance semantics to (sub)sets of $\mathcal{T}$; we simply say that $\mathcal{T}$ *θ-satisfies* τ, and denote it by:

$$\mathcal{T} \Vdash_\theta \tau.$$

Information-based learning. Propositional decision trees date back to Belson's [4] seminal work, based on which in [33] the authors proposed their innovative solution as an alternative to functional regression. The algorithm proposed in [32] is the first implementation of a decision tree for classification, but *CART* [8], *ID3* [36], and *C4.5* [37], are the most well-known. Because all share the same principles, they call can be generalized following the above schema, ending up in what we can generically call *information-based learning* of *temporal decision trees*. Information based learning is a general, greedy, sub-optimal approach to decision tree induction (optimal decision tree induction is knowingly NP-hard [22]). *Entropy-based learning* of (temporal) decision trees is a particular case of information-based learning, and the most common one. It works as follows. Let π_i be the fraction of instances labelled with class C_i in a dataset $\mathcal{T}$ with l distinct classes. Then, the *information conveyed* by $\mathcal{T}$ (or *entropy* of $\mathcal{T}$) is computed as:

$$Info(\mathcal{T}) = -\sum_{i=1}^{l} \pi_i \log \pi_i.$$

Intuitively, the entropy is inversely proportional to the purity degree of $\mathcal{T}$ with respect to the class values. In binary trees, *splitting*, which is the main greedy operation, is performed over a specific attribute A, a threshold value $a \in dom(A)$, a value γ, and the operators $\sim$ and $\bowtie$. Let $S(A, a, \gamma, \sim, \bowtie)$ be the decision entailed by $A, a, \gamma, \sim$, and $\bowtie$, and let $(\mathcal{T}_e, \mathcal{T}_u)$ be the partition of $\mathcal{T}$ entailed by $S(A, a, \gamma, \sim, \bowtie)$ (as defined above). The *splitting information* of $S = S(A, a, \gamma, \sim, \bowtie)$ is defined as:

$$InfoSplit(\mathcal{T}, S) = \frac{|\mathcal{T}_e|}{|\mathcal{T}|} Info(T_e) + \frac{|\mathcal{T}_u|}{|\mathcal{T}|} Info(T_u).$$

In this way, we can define the *entropy gain of a decision* as:

$$InfoGain(\mathcal{T}, S) = Info(\mathcal{T}) - InfoSplit(\mathcal{T}, S).$$

Existing open-source implementations of decision trees include, for example, the classes *DecisionTreeClassifier* in *Scikit-learn* [34] and *J48* in *WEKA* [40] learning frameworks, and the *DecisionTree* package [5] written in the *Julia* [6] programming language. In [38], an implementation of the algorithm for temporal decision tree learning based on the WEKA implementation of *C4.5*, that is, J48, and called *TemporalJ48* was presented. In recent years, the Julia programming language is becoming increasingly popular for scientific computing and, although the language is still young, there exists a stable Julia package for decision tree learning. Due to the performance gains that Julia, as a compiled language, enables, we developed an implementation of a temporal decision tree learning algorithm, called *TCART*, starting from the existing Julia package. Besides the language, *TCART* and *TemporalJ48* differ in implementation details only, and they can be considered the same algorithm to all intents and purposes.

Algorithm 1 High-level description of *TCART*.

```
function TCART(𝒯, n_att, n_lan):
    τ ← initialize an empty decision tree
    Preprocess(𝒯)
    root(τ) ← Learn(𝒯, n_att, n_lan)
    return τ
end
function Learn(𝒯, n_att, n_lan):
    if a stopping condition applies then return CreateLeafNode(𝒯)
    S ← FindBestDecision(𝒯, n_att, n_lan)      ◁ using n_att attr., n_lan modal op.
    (𝒯_e, 𝒯_u) ← Split(𝒯, S)
    ν ← CreateNode(𝒯)
    L(ν) ← Learn(𝒯_e)
    R(ν) ← Learn(𝒯_u)
    return ν
end
```

Algorithm 2 High-level description of *TRF*.

```
function TRF(𝒯, k, n_att, n_lan):
    Preprocess(𝒯)
    ℱ ← ∅
    foreach i ∈ [1, ..., k] do 𝒯' ← SubsetSample(𝒯, ⌈k/m⌉)      ◁ choose dataset
    τ ← TCART(𝒯', n_att, n_lan)
    ℱ ← ℱ ∪ {τ}
    return ℱ
end
```

Temporal random forests. In the propositional case, the generalization from single trees to forests of trees is relatively easy. The idea that underlies the so-called *random forests* model [7] is the following one: different trees can be learned from different subsets of the training set, using different subsets of attributes. Each tree is precisely a propositional decision tree; a *random forest classifier*, however, is a classifier whose instance semantics depends on many trees, and it is computed via some *voting* function. So, introducing temporal random forest models can be done in the same way. A *temporal random forest* is pair $(\mathcal{F}, v)$, where $\mathcal{F}$ is a collection of k temporal decision trees, that is, $\mathcal{F} = \{\tau_1, \ldots, \tau_k\}$, and $v : \mathcal{C}^k \to \mathcal{C}$ is *voting aggregation function* of all the unit votes of each temporal decision tree $\tau \in \mathcal{F}$.

Given a temporal random forest $(\mathcal{F} = \{\tau_1, \ldots, \tau_k\}, v)$ and a temporal instance $\mathcal{T} \in \mathcal{T}$, the *class assigned by* $\mathcal{F}$ *to* T, denoted by $\mathcal{F}(T)$, and called *instance semantics* of $\mathcal{F}$, is defined as:

$$v(\tau_1(T), \ldots, \tau_k(T)).$$

For a random forest $(\mathcal{F}, v)$ and a temporal data set $\mathcal{T}$, the notion $\mathcal{T} \Vdash_\theta \mathcal{F}$ is obtained, as in the case of the single tree, by the systematic application of the instance semantics to a certain (sub)set of the training dataset. Random forests differ from simple deterministic decision trees in many subtleties, all related to the learning algorithm. Such differences, along with the nature of the model, transform a purely symbolic method, such as decision trees, into a hybrid symbolic-functional approach. A first attempt towards random forests was made in [21], using the so-called *random subspace method*. Breiman's proposal [7], which can be considered the standard approach to random forests, was later introduced in the R learning package [27]. Julia [6] incorporates a class to generalize trees into forests [5]; we used such a class to create a *temporal random forest (TRF)* learning algorithm. *TRF* is based on a generalized version of *TCART* that allows one to use, at each step, only n_{att} attributes and n_{lan} modal operators to find the best split, as shown in Algorithm 1; as a matter of fact, this is a *randomized* version of the interval temporal logic decision tree learning strategy, which degenerates into the deterministic version when $n_{att} = n$ and $n_{lan} = 12$. This solution generalizes the propositional case of random forests in which, at each step of building a tree, only a subset of attributes is used as shown in the high-level description of *TRF* in Algorithm 2. In terms of implementation, both *TCART* and *TRF* need special attention to the supporting data structures. As a matter of fact, both the propositional and the temporal versions of the information-based decision tree learning algorithm run in polynomial time w.r.t. the size of the dataset, but the overhead introduced in the temporal case can be quite relevant, because of the high number of decisions that can be taken at each split. To solve this issue, the function *Preprocess* entails, among other steps, building a hash table keyed on the tuple $(T, [x, y], X, A, a, \sim)$ for specific values of γ, that returns the truth value of the decision $\langle X \rangle (A \bowtie_{\sim\gamma} a)$. In this way, at learning time, checking the information conveyed by a decision takes (virtual) constant time plus the time to compute the information function. Interestingly enough, such a structure is particularly useful for *TRF*: as a matter of fact, it can be computed beforehand and then shared by all instances of *TCART* without recomputing, effectively improving the overall experimental complexity with respect to k independent executions of *TCART*.

4 Data and Experiments

Data and preparation. Breath and cough open data gathered in [11] have the following structure. The entire dataset is composed by 9986 samples, recorded by 6613 volunteers. Out of all volunteers, 235 declared to be positive for COVID-19. The subjects were quasi-normally distributed by age, with an average between 30 and 39 and a frequency curve slightly left-skewed towards younger ones; the data is not gender-balanced, with more than double as many male subjects than female ones. Besides recording sound samples, subjects were asked to fill in a very small clinical history, plus information about their geographical location. The static data (that is, history and location) is used both here and in [11] to create specific datasets, called *tasks*, from the original one. In particular, the location of the subject has been used to distinguish among those that, at the moment of the recording, were living in almost-COVID-free countries; by combining this information with the subjects' declaration concerning a COVID-test, the negative subjects could be considered reliable. Of the three different tasks considered in [11], we focused on the first one, which is the problem distinguishing between subjects who were declared positive to COVID-19, from non-positive subjects with a *clean medical history*, who have *never smoked*, have *no symptoms*, and live in countries in which the virus spread at that moment was very low (and thus who can reliably be considered negative to the virus). As a result, this task counts 141 positive and 298 negative instances. In [11] the tasks were declined into nine versions, which differ by how subjects are represented, that is, using only their cough sample, only their breath sample, or both (giving rise to three different problems) and how data are preprocessed. Unfortunately, by treating breath and cough as time series, we can describe each instance with only one multivariate time series at the time; this is not a limit when using classical, static methods. Therefore, here we can approach only two of the above versions (referred to as *cough* version and *breath* version, respectively) of the problem. The raw audio data is encoded in the *Waveform Audio File (WAV)* format, and consists of a discrete sampling of the perceived sound pressure caused by (continuous) sound waves.

Despite the fact that this representation being already in the form of a time series, it is customary in audio signal processing to extract *spectral* representations of sounds, which facilitates their interpretation in terms of audio frequencies. To this end, we adopt a variation of a widespread representation technique, which goes under the name of *Mel-Frequency Cepstral Coefficients* (*MFCC*). MFCC, first proposed in [14], is still the preferred technique for extracting sensible spectral representations of audio data, and its use in machine learning has been fruitful for tackling hard AI tasks, such as speech recognition, music genre recognition, noise reduction, and audio similarity estimation. Computing the MFCC representation involves the following steps:

- **(i)** the raw audio is divided into (often overlapping) chunks of small size (e.g. $25ms$), and a *Discrete Fourier Transform (DFT)* is applied to each of the chunks, to produce a spectrogram of the sound at each chunk, that is, a continuous distribution of sound density across the frequency spectrum;
- **(ii)** the frequency spectrum is then warped according to a logarithmic function, which causes the frequency space to better reflect human ear perception of frequencies;
- **(iii)** a set of triangular band-pass filters is convolved across the frequency spectrum, discretizing it into a finite number of frequencies; finally,
- **(iv)** a *Discrete Cosine Transform (DCT)* is applied to the logarithm of the discretized spectrogram along the frequency axis, which compresses the spectral information at each point in time into a fixed number of coefficients.

This transformation does not modify the temporal ordering of the events; nevertheless, the classical approach at this point is to feed data to off-the-shelf classification methods which do not make use of such ordering (except, for example, recurrent neural networks). Moreover, the transformation does not preserve the spectral component, and the description of each time point is not directly related to sound frequencies. We applied MFCC up to step (*iii*), ultimately obtaining that each audio sample is represented as a time series with attributes describing the volume of different sound frequencies (called, here, *features*), and fed the resulting data to our temporal decision tree and random forest learning methods which are designed to learn natively from time series.

Test setting. We compare a temporal decision tree model (*TDT*), and two temporal random forest models with 50 and 100 trees, respectively. As for the random forest models, different choices for sampling the random subspace of decisions were explored:

- **(i)** considering the decisional space in its entirety;
- **(ii)** subsampling a low number of randomly-chosen features;
- **(iii)** subsampling a low number of randomly-chosen relations;
- **(iv)** subsampling a low number of randomly-chosen of features and relations.

After an initial pre-screening, we concluded that for this particular problem the best results are obtained by using either all relations and all features, or all relations and half of the features. While single decision trees experiments have been run with the standard pre-pruning setting (minimum entropy gain of 0.01 for a split to be meaningful, and a maximum entropy at each leaf of 0.6), random forest grow full trees.

As for the data, the chunk size and overlap for the DFT were fixed to the standard values of $15ms$ and $25ms$, respectively, and, since the processed series in this form present many points (100 for each second of recording), a moving average filter was applied, and the resulting series were capped at a maximum of 30 time points. Ultimately, different parametrizations were investigated by varying the number (f) of filters (frequencies), the size (w) and step (s) of the moving average, and by performing/not performing a *peak normalization* step prior to the MFCC phase. After the pre-screening phase, the moving average size and step have been fixed to 75 and 50 for the cough version, and to 45 and 30 for the breath version; moreover, $f \in \{40, 60\}$ for cough and $f \in \{20, 40\}$ for breath. In all cases we let $\bowtie$ be in $\{\leq, \geq\}$, $\sim$ be $\geq$, and $\gamma = 0.8$. Both versions count 439 audio samples, of which, as recalled above, 141 are positives and 298 are negatives; to minimize the bias, the dataset is, first, balanced by downsampling the majority class (thus deriving a subset of 282 samples), and, second, split into two (balanced) sets for training (80%) and test (20%). Since these two steps are randomized, this process is repeated 5 times (with seeds from 1 to 5), and we show the results of each repetition plus their average and standard deviation. Moreover, for a fixed training and test set, TRF is run 5 times with different seeds (again, with seeds from 1 to 5), and only their average is shown; so, for example, the first line in Tab. 2, is itself the result of averaging five runs on the same training/test splitting.

Results. The following questions are interesting: Is temporal random forest a suitable method to solve this problem? Which combination of parameters gives the best results? Are our best results comparable with the results obtained by standard techniques, especially in [11]? How our results can be interpreted?

As much as the suitability of our method is concerned, let us focus on Tab. 2, first, in which we show the detailed results for the case of temporal random forests; performances are all expressed in percentage. Each group of results is characterized by some parameters, and in particular different groups correspond to different numbers of features (20,40, or 60), the fact that cough or breath samples were used, and the fact that peak normalization was used (N), or not. The first general observation is that the average accuracy obtained by describing the subjects with their cough sample is 72.58 with a standard deviation of 0.72, while it is 69.40 with a standard deviation of 2.16 when subjects are described by their breath sample; this means that cough samples have a clear distinguishing power which is captured by temporal random forests. The standard *t-test* run on the populations of accuracies (averaged over the five runs) results in the two cases gives a *p-value* < 0.001, indicating that cough is clearly more informative than breath, for this particular task. In terms of sensitivity versus specificity, the best performance is obtained by a forest, learned on cough samples, with 100 trees and 60 features, using half of them in each tree: 71.57 and 74.86, respectively. This means that this model is able to correctly identify a real COVID-19-positive subject from his/her cough sample almost than 3 out of 4 times, with less than 28% of false negative cases. This particular configuration showed also a peak in performance during the first run (sensitivity: 81.43), possibly indicating that some samples are more informative than others. Comparing our results with [11] is not immediate: we could not use instances described by both cough and breath samples (and this combination resulted, in [11], more performing than using cough or breath only), we used a moving window technique for data preparation, and, more importantly, we limited our samples to the first 30 points of each sample, which corresponds to a length between 5 and 15 seconds (while the original samples have a length of around 30 seconds in most cases). Taking all this into account, however, we perform better than [11]: our best model improves the best model from [11] for the same data (task 1) by 2% in both precision and sensitivity.

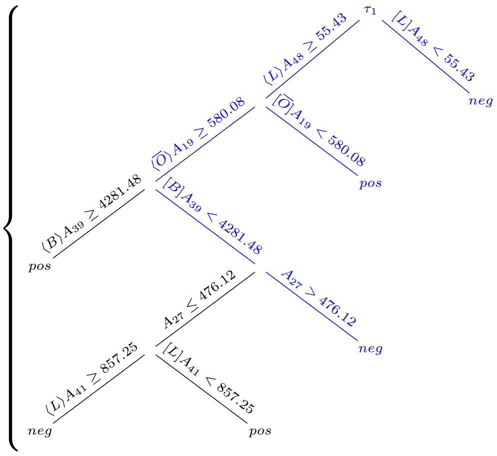
Table 2 Detailed results: random forests.

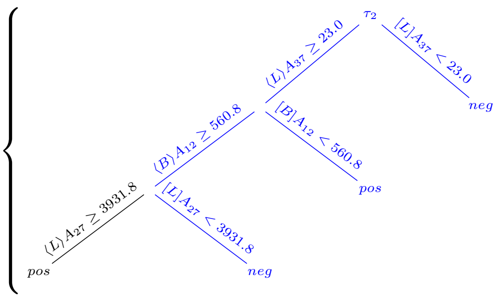
TRF		50 trees, all features				50 trees, half features				100 trees, all features				100 trees, half features			
		sens	spec	prec	acc	sens	spec	prec	acc	sens	spec	prec	acc	sens	spec	prec	acc
cough,40,N	1	80.0	85.0	84.53	82.5	78.57	83.57	82.93	81.07	79.29	88.57	87.77	83.93	80.0	85.0	84.3	82.5
	2	75.71	61.43	66.4	68.57	75.0	57.86	64.04	66.43	73.57	63.57	66.89	68.57	74.29	59.29	64.65	66.79
	3	58.57	78.57	73.44	68.57	55.0	81.43	74.65	68.21	57.86	79.29	73.83	68.57	57.86	80.71	75.15	69.29
	4	66.43	75.0	72.69	70.71	70.0	75.0	73.72	72.5	71.43	74.29	73.58	72.86	70.71	75.71	74.51	73.21
	5	65.0	68.57	67.42	66.79	70.71	77.14	75.69	73.93	68.57	69.29	69.08	68.93	69.29	76.43	74.65	72.86
	avg	**69.14**	**73.71**	**72.9**	**71.43**	**69.86**	**75.0**	**74.21**	**72.43**	**70.14**	**75.0**	**74.23**	**72.57**	**70.43**	**75.43**	**74.65**	**72.93**
	std	8.62	9.08	7.21	6.34	9.0	10.16	6.75	5.72	7.91	9.57	8.13	6.6	8.15	9.76	6.95	5.97
cough,60,N	1	79.29	83.57	82.94	81.43	80.71	83.57	83.19	82.14	80.71	84.29	83.69	82.5	78.57	83.57	82.9	81.07
	2	79.29	57.86	65.39	68.57	74.29	62.14	66.21	68.21	77.14	60.71	66.33	68.93	78.57	63.57	68.35	71.07
	3	57.86	78.57	73.18	68.21	55.0	77.86	71.42	66.43	57.14	80.71	74.76	68.93	58.57	80.71	75.36	69.64
	4	66.43	75.0	72.87	70.71	68.57	74.29	72.74	71.43	70.0	75.0	73.96	72.5	67.86	75.71	73.63	71.79
	5	67.86	70.0	69.36	68.93	69.29	73.57	72.49	71.43	66.43	68.57	67.89	67.5	69.29	74.29	73.02	71.79
	avg	**70.15**	**73.0**	**72.75**	**71.57**	**69.57**	**74.29**	**73.21**	**71.93**	**70.28**	**73.86**	**73.33**	**72.07**	**70.57**	**75.57**	**74.65**	**73.07**
	std	9.18	9.81	6.51	5.6	9.48	7.86	6.18	6.1	9.27	9.46	6.86	6.12	8.38	7.68	5.29	4.56
cough,40	1	76.43	80.0	79.89	78.21	73.57	74.29	74.15	73.93	75.0	80.71	79.67	77.86	77.14	80.0	79.49	78.57
	2	70.0	66.43	67.73	68.21	70.0	63.57	65.81	66.79	69.29	65.0	66.59	67.14	70.71	62.86	65.56	66.79
	3	59.29	84.29	79.15	71.79	57.14	82.86	77.36	70.0	57.14	84.29	78.55	70.71	58.57	87.14	82.14	72.86
	4	74.29	72.86	73.29	73.57	73.57	71.43	71.98	72.5	74.29	75.71	75.38	75.0	74.29	75.0	74.83	74.64
	5	70.71	81.43	79.27	76.07	72.14	78.57	77.25	75.36	71.43	77.14	75.87	74.29	72.14	80.0	78.33	76.07
	avg	**70.14**	**77.0**	**75.87**	**73.57**	**69.28**	**74.14**	**73.31**	**71.72**	**69.43**	**76.57**	**75.21**	**73.0**	**70.57**	**77.0**	**76.07**	**73.79**
	std	6.61	7.26	5.28	3.86	6.94	7.33	4.76	3.39	7.24	7.27	5.14	4.15	7.13	9.01	6.44	4.43
cough,60	1	75.71	78.57	77.94	77.14	75.0	76.43	76.16	75.71	76.43	77.86	77.65	77.14	81.43	76.43	77.63	78.93
	2	70.0	63.57	65.96	66.79	74.29	63.57	67.15	68.93	75.0	60.0	65.21	67.5	72.14	60.71	64.73	66.43
	3	54.29	81.43	74.56	67.86	57.86	83.57	78.08	70.71	56.43	80.71	74.52	68.57	59.29	80.0	74.95	69.64
	4	72.14	72.86	72.74	72.5	73.57	77.14	76.48	75.36	74.29	73.57	73.78	73.93	72.14	75.71	74.88	73.93
	5	72.14	79.29	77.85	75.71	70.71	80.0	77.88	75.36	72.86	80.71	79.16	76.79	72.86	81.43	79.7	77.14
	avg	**68.86**	**75.14**	**73.81**	**72.0**	**70.29**	**76.14**	**75.15**	**73.21**	**71.0**	**74.57**	**74.06**	**72.79**	**71.57**	**74.86**	**74.38**	**73.21**
	std	8.4	7.2	4.92	4.6	7.13	7.57	4.55	3.16	8.25	8.65	5.42	4.53	7.91	8.26	5.76	5.18
breath,20,N	1	77.14	68.57	71.65	72.86	77.86	69.29	71.92	73.57	75.71	71.43	72.66	73.57	77.14	68.57	71.22	72.86
	2	60.71	58.57	59.33	59.64	62.86	54.29	57.88	58.57	60.71	52.86	56.34	56.79	59.29	54.29	56.49	56.79
	3	60.0	75.0	70.95	67.5	63.57	76.43	73.68	70.0	65.0	70.0	68.5	67.5	62.86	76.43	73.07	69.64
	4	64.29	65.71	65.25	65.0	62.86	68.57	66.7	65.71	64.29	67.14	66.17	65.71	64.29	69.29	67.68	66.79
	5	70.71	80.71	78.99	75.71	67.14	75.71	73.57	71.43	72.14	77.86	76.98	75.0	72.14	76.43	75.52	74.29
	avg	66.57	69.71	69.23	68.14	66.86	68.86	68.75	67.86	67.57	67.86	68.13	67.71	67.14	69.0	68.8	68.07
	std	7.27	8.52	7.38	6.37	6.4	8.9	6.71	5.93	6.16	9.26	7.78	7.26	7.3	9.04	7.45	6.95
breath,40,N	1	75.71	58.57	64.75	67.14	72.86	64.29	67.19	68.57	77.86	60.71	66.54	69.29	74.29	62.14	66.28	68.21
	2	66.43	55.71	60.02	61.07	62.86	52.14	56.83	57.5	64.29	55.71	59.33	60.0	60.0	53.57	56.39	56.79
	3	62.86	73.57	70.46	68.21	60.0	73.57	70.07	66.79	63.57	70.71	68.5	67.14	62.14	70.71	67.99	66.43
	4	62.14	68.57	66.54	65.36	62.14	65.0	64.02	63.57	60.71	69.29	66.46	65.0	64.29	66.43	65.81	65.36
	5	68.57	81.43	78.65	75.0	69.29	80.0	77.73	74.64	70.0	80.71	78.75	75.36	70.0	81.43	79.2	75.71
	avg	67.14	67.57	68.08	67.36	65.43	67.0	67.17	66.21	67.29	67.43	67.92	67.36	66.14	66.86	67.13	66.5
	std	5.46	10.62	7.0	5.07	5.41	10.54	7.69	6.32	6.8	9.66	6.99	5.65	5.88	10.32	8.13	6.77
breath,20	1	72.14	77.86	76.59	75.0	70.71	75.71	74.54	73.21	72.86	77.86	76.72	75.36	72.14	76.43	75.45	74.29
	2	65.0	72.14	70.0	68.57	70.71	66.43	67.61	68.57	65.71	70.71	69.21	68.21	73.57	66.43	68.61	70.0
	3	65.0	70.0	68.48	67.5	62.86	70.71	68.35	66.79	68.57	70.71	70.18	69.64	68.57	68.57	68.56	68.57
	4	62.14	78.57	74.65	70.36	64.29	77.86	74.92	71.07	65.0	80.0	76.5	72.5	65.71	77.14	74.38	71.43
	5	62.86	83.57	79.52	73.21	67.86	82.86	80.26	75.36	62.86	85.71	81.63	74.29	66.43	86.43	83.32	76.43
	avg	65.43	76.43	73.85	70.93	67.29	74.71	73.14	71.0	67.0	77.0	74.85	72.0	**69.28**	**75.0**	**74.06**	**72.14**
	std	3.96	5.42	4.58	3.14	3.62	6.36	5.23	3.45	3.86	6.42	5.14	3.03	3.46	7.94	6.08	3.2
breath,40	1	69.29	80.0	77.73	74.64	71.43	79.29	77.79	75.36	71.43	77.86	76.47	74.64	72.14	77.86	76.54	75.0
	2	61.43	72.86	69.34	67.14	66.43	70.71	69.39	68.57	62.86	70.71	68.14	66.79	67.14	68.57	68.05	67.86
	3	65.0	71.43	69.35	68.21	65.0	69.29	67.86	67.14	67.86	68.57	68.32	68.21	67.86	67.86	67.85	67.86
	4	62.86	80.0	76.05	71.43	62.14	78.57	74.54	70.36	62.14	80.0	75.65	71.07	62.86	80.0	75.89	71.43
	5	62.14	87.86	84.06	75.0	64.29	85.0	81.23	74.64	64.29	88.57	85.43	76.43	66.43	87.86	84.83	77.14
	avg	64.14	78.43	75.31	71.28	65.86	76.57	74.16	71.21	65.72	77.14	74.8	71.43	67.29	76.43	74.63	71.86
	std	3.17	6.59	6.21	3.6	3.48	6.52	5.61	3.65	3.88	7.97	7.12	4.1	3.33	8.38	7.05	4.18

Single decision trees, whose results are shown in Tab. 3, are not as performing as forests of trees: the maximum averaged (only on different training/test splits: single tree learning is a deterministic process) accuracy reached with a single tree is 66%, which is less than the minimum accuracy reached with forests of trees. But single trees can be interpreted. In Tab. 3, right-hand side, each displayed tree has been learned in the conditions indicated by the arrow (so, for example, the topmost one has been learned by cough samples, with 60 features, no normalization, and with the subsampling obtained by seed 1), but in full training mode; in full training mode, the difference between the five runs of each configuration is limited to the downsampling phase. On each of the trees, we selected specific leaves with high *confidence* and *support* (in Tab. 3, right-hand side, these correspond to highlighted branches).

Table 3 Detailed results: single temporal decision trees.

TDT		sens	spec	prec	acc
cough,40,N	1	64.29	71.43	69.23	67.86
	2	64.29	35.71	50.0	50.0
	3	57.14	67.86	64.0	62.5
	4	67.86	50.0	57.58	58.93
	5	64.29	64.29	64.29	64.29
	avg	63.57	57.86	61.02	60.72
	std	3.91	14.81	7.42	6.8
cough,60,N	1	85.71	75.0	77.42	80.36
	2	71.43	39.29	54.05	55.36
	3	57.14	82.14	76.19	69.64
	4	57.14	67.86	64.0	62.5
	5	64.29	60.71	62.07	62.5
	avg	67.14	65.0	66.75	66.07
	std	11.95	16.44	9.92	9.45
cough,40	1	64.29	75.0	72.0	69.64
	2	71.43	50.0	58.82	60.71
	3	35.71	89.29	76.92	62.5
	4	60.71	50.0	54.84	55.36
	5	71.43	64.29	66.67	67.86
	avg	60.71	65.72	65.85	63.21
	std	14.73	16.87	9.11	5.73
cough,60	1	67.86	75.0	73.08	71.43
	2	71.43	46.43	57.14	58.93
	3	50.0	78.57	70.0	64.29
	4	60.71	53.57	56.67	57.14
	5	67.86	60.71	63.33	64.29
	avg	63.57	62.86	64.04	63.22
	std	8.53	13.74	7.41	5.59
breath,20,N	1	60.71	71.43	68.0	66.07
	2	60.71	50.0	54.84	55.36
	3	46.43	75.0	65.0	60.71
	4	53.57	53.57	53.57	53.57
	5	57.14	75.0	69.57	66.07
	avg	55.71	65.0	62.2	60.36
	std	5.97	12.22	7.49	5.84
breath,40,N	1	64.29	67.86	66.67	66.07
	2	64.29	39.29	51.43	51.79
	3	53.57	71.43	65.22	62.5
	4	64.29	57.14	60.0	60.71
	5	57.14	78.57	72.73	67.86
	avg	60.72	62.86	63.21	61.79
	std	5.05	15.28	8.0	6.26
breath,20	1	71.43	39.29	54.05	55.36
	2	78.57	39.29	56.41	58.93
	3	71.43	60.71	64.52	66.07
	4	64.29	75.0	72.0	69.64
	5	60.71	78.57	73.91	69.64
	avg	69.29	58.57	64.18	63.93
	std	6.96	18.83	8.93	6.49
breath,40	1	75.0	53.57	61.76	64.29
	2	78.57	35.71	55.0	57.14
	3	57.14	50.0	53.33	53.57
	4	71.43	71.43	71.43	71.43
	5	67.86	75.0	73.08	71.43
	avg	70.0	57.14	62.92	63.57
	std	8.22	16.17	9.11	8.15





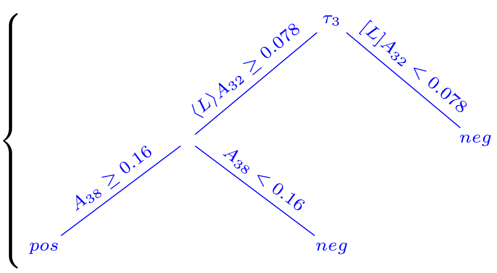


Table 4 Examples of extracted rules; all constants are scaled by a factor 10^3, and the symbols $\geq$ and $\leq$ (resp., their duals $<$ and $>$) denote $\geq_{\geq 0.8}$ and $\leq_{\geq 0.8}$ (resp., $<_{\geq 0.2}$ and $>_{\geq 0.2}$).

tree	rule	conf.	sup.
tree τ_1	$[L]A_{48} < 55.43 \Rightarrow neg$	0.74	0.37
	$\langle L\rangle A_{48} \geq 55.43 \wedge [L](A_{48} \geq 55.43 \rightarrow \overline{[O]}A_{19} < 580.08) \Rightarrow pos$	0.86	0.31
	$\langle L\rangle(A_{48} \geq 55.43 \wedge \overline{\langle O\rangle}A_{19} > 580.08 \wedge \overline{[O]}(A_{19} > 580.08 \rightarrow [B](A_{39} < 4281.48 \wedge A_{27} > 476.12))) \Rightarrow neg$	0.89	0.16
tree τ_2	$[L]A_{37} < 23.0 \Rightarrow neg$	0.73	0.40
	$\langle L\rangle A_{37} \geq 23.0 \wedge [L](A_{37} \geq 23.0 \rightarrow [B]A_{12} < 560.8) \Rightarrow pos$	0.83	0.39
	$\langle L\rangle(A_{37} \geq 23.0 \wedge \langle B\rangle(A_{12} \geq 560.8 \wedge [B](A_{12} \geq 560.8 \rightarrow [L]A_{27} < 3931.8))) \Rightarrow neg$	0.83	0.17
tree τ_3	$[L]A_{32} < 0.078 \Rightarrow neg$	0.77	0.36
	$\langle L\rangle A_{32} \geq 0.078 \wedge [L](A_{32} \geq 0.078 \rightarrow A_{38} < 0.16) \Rightarrow neg$	0.78	0.14
	$\langle L\rangle(A_{32} \geq 0.078 \wedge A_{38} \geq 0.016) \Rightarrow pos$	0.77	0.5

Each of these leaves can be interpreted as a classification rule of the type $\varphi \Rightarrow pos/neg$, where φ if the path-formula that can be extracted from the branch as explained in Section 3. The result of such an interpretation is in Tab. 4, in which every rule has been made explicit and its confidence and support displayed. Rules of this type can be visualized by synthesizing their model (in logical terms), and even converted into audible sound, which corresponds to *what should be heard (in a cough sample) to suspect a COVID-19 infection*. Obviously, implementing a tool for real-time screening based on rules as simple as these ones is much easier and direct than performing complex higher-dimensional matrix computations.

5 Conclusions

The ability of explaining the underlying theory that is extracted with machine learning methods is of uttermost importance, especially in medicine applications. Interpretability and explainability in learning are often synonym of a symbolic approach, which, in turn, should be based on logics that are able to capture the complexity of the phenomena. Modal symbolic learning offers classical learning tools enhanced with modal propositional logics that allow one to extract complex information from data; temporal symbolic learning is the specialization of modal symbolic learning to the case of temporal data and temporal logics. In the recent literature temporal decision trees, based on Halpen and Shoham's interval temporal logic HS have been proposed for learning from multivariate time series. In this paper we proposed a generalization of temporal decision trees to temporal random forest, following the path traced in the propositional case. In order to test our method, we applied it to the case of recognizing COVID-19-positive subjects from negative ones using a recording of cough/breath sample, interpreted as a multivariate time series. Not only this approach is completely innovative, but our performances are superior to those of classical methodologies applied to the same data, while allowing the interpretation of the results, and enabling the visualization, and even the transformation in audible sounds of the models that represent the distinguishing characteristics of a cough/breath sample of a positive subject. In abstract terms, such an ability could be useful to train medical personnel to recognize positive subjects, but also to develop automatic procedures that perform a (rough) screening, for example as a smartphone application.

This work is part of a larger project that aims to generalize symbolic learning methods with modal logics in a systematic way. As we have seen, not only this presents new challenges at the mathematical level, but also at the implementation one. Open problems at this moment include studying better interpretation techniques for temporal, and, in general, modal random forests, that do not require resorting to single trees, studying more efficient data structures that require less computational power, both in terms of space and time, for symbolic learning, and completing the generalization of the whole range of symbolic methods, from decision trees to rule based classifiers, with deterministic and randomized learning methods, and with underlying logics both crisp and fuzzy. Exploring how to learn from *multi-frame* dimensional data is also on the agenda; this would allow us to improve our performances in the classification between COVID/non-COVID subjects, as we would be able to describe them using both their cough and breath samples. In the end, we want to test our methodologies on all three tasks from [11], to establish if there is an improvement in all cases, and if the rules that we are able to extract are indeed clinically useful in the fight against the virus.

References

1 J.F. Allen. Maintaining knowledge about temporal intervals. *Communication of the ACM*, 26(11):832–843, 1983. doi:10.1145/182.358434.

2 S. Balakrishnan and D. Madigan. Decision trees for functional variables. In *Proc. of the 6th International Conference on Data Mining*, pages 798–802, 2006.

3 E. Bartocci, L. Bortolussi, and G. Sanguinetti. Data-driven statistical learning of temporal logic properties. In *Proc. of the 12th International Conference on Formal Modeling and Analysis of Timed Systems*, volume 8711 of *Lecture Notes in Computer Science*, pages 23–37. Springer, 2014.

4 W.A. Belson. A technique for studying the effects of television broadcast. *Journal of the Royal Statistical Society*, 5(3):195–202, 1956.

5 J. Bezanson, A. Edelman, S. Karpinski, and V.B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

6 J. Bezanson, A. Edelman, S. Karpinski, and V.B. Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

7 L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

8 L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth Publishing Company, 1984.

9 D. Bresolin, D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco. Decidability and complexity of the fragments of the modal logic of Allen's relations over the rationals. *Information and Computation*, 266:97–125, 2019.

10 D. Bresolin, D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco. Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. *Theoretical Computers Science*, 560:269–291, 2014.

11 C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo. Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data. In *Proc. of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3474–3484, 2020.

12 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.

13 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 Intelligences*, volume 11468 of *Lecture Notes in Computer Science*, pages 778–793. Springer, 2019.

14 S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

15 J.J. Rodríguez Diez, C. Alonso González, and H. Boström. Boosting interval based literals. *Intelligent Data Analysis*, 5(3):245–262, 2001.

16 H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

17 J.H. Friedman and B.E. Popescu. Predictive learning via rule esambles. *The Annals of Applied Statistics*, 2(3), 2008.

18 V. Goranko, A. Montanari, and G. Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1-2):9–54, 2004.

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

20 J. Han, K. Qian, M. Song, Z. Yang, Z. Ren, S. Liu, J. Liu, H. Zheng, W. Ji, T. Koike, X. Li, Z. Zhang, Y. Yamamoto, and B. Schuller. An early study on intelligent analysis of speech under covid-19: Severity, sleep quality, fatigue, and anxiety. In *Proc. of the Conference INTERSPEECH*, pages 1–5, 2020.

21 T.K. Ho. Random decision forests. In *Proc. of the 3rd International Conference on Document Analysis and Recognition*, pages 278–282, 1995.

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

23 A. Imran, I. Posokhova, H.N. Qureshi, U. Masood, M. Sajid Riaz, K. Ali, C.N. John, I. Hussain, and M. Nabeel. AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. *Informatics in Medicine Unlocked*, 20:1–14, 2020.

24 Y. Kakizawa, R.H. Shumway, and M. Taniguchi. Discrimination and clustering for multivariate time series. *Journal of the American Statistical Association*, 93(441):328–340, 1998.

25 M. Kudo, J. Toyama, and M. Shimbo. Multidimensional curve classification using Passing-Through regions. *Pattern Recognition Letters*, 20(11):1103–1111, 1999.

26 J. Laguarta, F. Hueto, and B. Subirana. COVID-19 artificial intelligence diagnosis using only cough recordings. *IEEE Open Journal of Engineering in Medicine and Biology*, 1:275–281, 2020.

27 A. Liaw and M. Wiener. Classification and regression by RandomForest. *R News*, 2(3):18–22, 2002.

28 J. Lines and A.J. Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592, 2015.

29 E. Lucena-Sánchez and and I.E. Stan G. Sciavicco. Feature and language selection in temporal symbolic regression for interpretable air quality modelling. *Algorithms*, 14(3):1–17, 2021.

30 P. Malhotra, V. TV, L. Vig, P. Agarwal, and G. M. Shroff. Timenet: Pre-trained deep recurrent neural network for time series classification. In *Proc. of the 25th European Symposium on Artificial Neural Networks*, pages 607–612, 2017.

31 N. Meinshausen. Node harvest. *The Annals of Applied Statistics*, 4(4), 2010.

32 R. Messenger and L. Mandell. A modal search technique for predictive nominal scale multivariate analysis. *Journal of the American Statistical Association*, 67(340):768–772, 1972.

33 J.N. Morgan and J.A. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of American Statistical Association*, 58(302):415–434, 1963.

34 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

35 R.X.A. Pramono, S. Bowyer, and E. Rodriguez-Villegas. Automatic adventitious respiratory sound analysis: A systematic review. *Plos One*, 12(5):1–43, 2017.

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

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

38 G. Sciavicco and I.E. Stan. Knowledge Extraction with Interval Temporal Logic Decision Trees. In *Proc. of the 27th International Symposium on Temporal Representation and Reasoning*, volume 178 of *Leibniz International Proceedings in Informatics*, pages 9:1–9:16, 2020.

39 I. Sutskever, O. Vinyals, and Q.V. Le. Sequence to sequence learning with neural networks. In *Proc. of the 28th Conference on Neural Information Processing Systems*, pages 3104–3112, 2014.

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

41 Y. Yamada, E. Suzuki, H. Yokoi, and K. Takabayashi. Decision-tree induction from time-series data based on a standard-example split test. In *Proc. of the 12th International Conference on Machine Learning*, page 840–847. AAAI Press, 2003.