Artificial Intelligence In Medicine 137 (2023) 102486

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Research paper

The voice of COVID-19: Breath and cough recording classification with temporal decision trees and random forests

F. Manzella, G. Pagliarini, G. Sciavicco [*], I.E. Stan

Department of Mathematics and Computer Science, University of Ferrara, Italy

ARTICLE INFO

MSC:
68T27
62P10

Keywords:
Interval temporal decision trees and forests
Symbolic models
COVID-19

ABSTRACT

Symbolic learning is the logic-based approach to machine learning, and its mission is to provide algorithms and methodologies to extract logical information from data and express it in an interpretable way. 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. In order to improve their performances, interval temporal decision trees can be embedded into interval temporal random forests, mimicking the corresponding schema at the propositional level. In this article we consider a dataset of cough and breath sample recordings of volunteer subjects, labeled with their COVID-19 status, originally collected by the University of Cambridge. By interpreting such recordings as multivariate time series, we study the problem of their automated classification using interval temporal decision trees and forests. While this problem has been approached with the same dataset as well as with other datasets, in all cases, non-symbolic learning methods (usually, deep learning-based) have been applied to solve it; in this article we apply a symbolic approach, and show that it does not only outperform the state-of-the-art obtained with the same dataset, but its results are also superior to those of most non-symbolic techniques applied on other datasets. As an added bonus, thanks to the symbolic nature of our approach, we are also able to extract explicit knowledge to help physicians characterize typical COVID-positive cough and breath.

1. Introduction

Machine learning. Machine Learning (ML) is at the core of modern Artificial Intelligence (AI). 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, up to recognizing voices, signatures, images, and many others. On one hand, ML methods can be classified as *parametric* and *nonparametric*. Parametric methods attempt to learn models in terms of a fixed-size set of parameters, and are, therefore, based on strong assumptions on the *structure* of the function to be learnt; typical examples are *linear/logistic regression models*, *naive bayes*, and *neural networks*, where the size of the model is fixed prior to the learning phase. On the contrary, nonparametric methods do not make structural assumptions, and their complexity is, instead, driven by the complexity of the process that is to be seized; examples are *k-nearest neighbors*, *support vector machines*, and *decision trees*. On the other hand, ML techniques can be also regarded as *functional* or *symbolic*. A functional learning method learns an *algebraic function* that represents the 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; representative symbolic learning models are *decision trees* and *rule-based classifiers*, which also happen to be nonparametric. Symbolic learning is sometimes statistically less accurate than functional one, as it based on representing coarse concepts in numerical domains, but its results can be interpreted and explained by humans, while functional models are generally 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-static data were usually dealt with propositional methods by first *flattening* them using global features (e.g., instead of considering the raw evolution of a patient's temperature within the monitored period, consider only the average temperature value). These *lossy* procedures allow using off-the-shelf methods, but severely hampers the interpretability of the results, and, in many cases, the statistical performances of the extracted model as well. *Interval temporal*

[*] Corresponding author.
E-mail addresses: federic.manzella@edu.unife.it (F. Manzella), giovanni.pagliarini@unife.it (G. Pagliarini), guido.sciavicco@unife.it (G. Sciavicco), ioneleduard.stan@unife.it (I.E. Stan).

https://doi.org/10.1016/j.artmed.2022.102486
Received 6 May 2022; Received in revised form 27 December 2022; Accepted 28 December 2022
Available online 4 February 2023

Table 1
Overview of the existing work in on COVID-19 diagnosis from audio recordings of *cough* (c), *breath* (b), or *speech* (s). Only works framing the problem as a binary classification one are listed, and for each work the type of model deployed, the dataset, the model evaluation setting, and an indication of the best classification performance attained is shown. For the datasets, CAM stands for *Cambridge* [8], V for *Virufy* [9], CO for *COUGHVID* [10], CW for *Coswara* [11], NO for *NoCoCoDa* [12], C19S for *COVID-19 Sounds* [13], DI for *DiCOVA* [14], CCS and CSS for *ComParE2021 CCS* and *CSS* [15]. As for evaluation setting, CV stands for *cross-validation*, while LOOCV for *leave one out cross-validation*. As for performance evaluation, AUC stands for *area under the ROC curve*, ACC for *accuracy*, and UAR for *unweighted average recall*.

	Work	c	b	s	Model type	Dataset	Setting	Best
2020	Brown et al. [8]	✓	✓		LR, SVM	CAM	10-fold CV	AUC: 88
	Imran et al. [16]		✓		CNN	own	5-fold CV	ACC: 92
	Hassan et al. [17]	✓	✓	✓	LSTM	own	train/test	ACC: 98
	Laguarta et al. [18]	✓			CNN	own	train/test	AUC: 97
	Chaudhari et al. [9]	✓		✓	CNN	V, CO, CW	train/test	AUC: 77
	Bansal et al. [19]	✓			CNN	own	train/test	AUC: 71
2021	Melek [20]	✓			k-NN	V, NO	LOOCV	ACC: 98
	Xia et al. [21]	✓	✓	✓	CNN	own	10-fold CV	AUC: 74
	Pahar et al. [22]	✓			CNN, LSTM	own, CW	5-fold CV	ACC: 95
	Despotovic et al. [23]	✓	✓	✓	CNN, RF	own	5-fold CV	ACC: 89
	Dash et al. [24]	✓	✓	✓	SVM	CW, CAM	5-fold CV	ACC: 85
	Stasak et al. [25]			✓	DT	own	5-fold CV	ACC: 80
	Han et al. [26]			✓	SVM	C19S	5-fold CV	ACC: 79
	Muguli et al. [14]	✓	✓	✓	LR, RF	DI	5-fold CV	AUC: 75
	Coppock et al. [27]	✓	✓		CNN	CAM	3-fold CV	AUC: 91
	Fakhry et al. [28]	✓			CNN	CO	holdout	AUC: 99
	Das et al. [29]	✓			LR, RF	DI	holdout	AUC: 81
	Xia et al. [13]	✓	✓	✓	SVM, CNN	own	holdout	AUC: 75
	Casanova et al. [30]	✓		✓	CNN	CCS, CSS	holdout	UAR: 76
	Schuller et al. [15]	✓		✓	SVM	own	holdout	UAR: 74
	Deshpande and Schuller [31]	✓			LSTM	DI	train/test	AUC: 64
2022	Alkhodari et al. [32]		✓		CNN, LSTM	CW	LOOCV	ACC: 95
	Dentamaro et al. [33]	✓	✓		CNN	CAM, CW	10-fold CV	ACC: 93
	Tena et al. [34]	✓			RF	own	10-fold CV	ACC: 90
	Chang et al. [35]	✓			CNN	CO	5-fold CV	ACC: 72
	Nassif et al. [36]	✓	✓	✓	CNN, LSTM	own	holdout	ACC: 98
	Aly et al. [37]	✓	✓	✓	NN	CW	holdout	ACC: 96
	Han et al. [38]	✓	✓	✓	CNN	C19S	holdout	AUC: 71

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 [1,2], and have shown great potential as a method to classify multivariate time series. Propositional decision trees can be generalized into *random forests* [3] 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 to be 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, it is still possible [4–6]. Building on this idea, *interval temporal logic random forests* can be used to improve the performances of single interval temporal decision trees. Interval temporal forests follow the same principles as the propositional ones: a forest is a schema based on the idea of training different independent trees on different samples and different attributes; in the temporal case, moreover, they may also be trained on different interval relations (in [7], the problem of selecting subsets of relations in the learning phase, treated as feature selection problem, has been studied).

COVID-19. *COVID-19* is a respiratory disease caused by the *SARS-CoV2* virus. The disease was classified in 2019, and caused a pandemic that occurred during the years 2020, 2021, and is still partially ongoing in 2022. The current scientific literature on COVID-19 is immense, and it ranges everywhere from medicine to economy, sociology, psychology, among many other fields. AI is no exception: as of April 2022, a simple Scopus search for `COVID-19 AND Artificial Intelligence` returns about 4000 entries. Perhaps one of the most appealing lines of research, in this regard, deals with the possibility of deriving computational models for the diagnosis of COVID-19 from respiratory sounds of human subjects, an idea already largely explored for diagnosing other respiratory diseases such as bronchitis or pneumonia. Diagnosis is usually enabled via coughs, breaths and oral speech audio samples, which can be easily recorded with smartphone hardware. This has a double advantage: it eases the availability of data, which can be quickly crowdsourced throughout smartphone applications, and it facilitates even more the actual diagnosis, which can be performed in real-time by the same applications. Together, the whole process provides a compelling, zero-cost, non-invasive alternative to the widely used diagnostic (e.g., antigen or molecular) tests.

Artificial intelligence and COVID-19. Table 1 lists relevant works published in 2020–2022 that attempts at the task of COVID-19 diagnosis via audio samples of cough, breaths, and/or speech, in various combinations. While different models are not easily comparable, not even in performances, this review reveals a unmistakable trend towards using functional methods for this task (with the exception of [25], where decision tree models are used), and, in fact, the concepts of transparency and interpretability of the models are not even mentioned in these contributions. This is in sharp contrast with the need of understanding the reasons that underly the decisions taken by intelligent systems, and in particular those related to medical diagnoses; such a need is widely shared in the community, and witnessed, for example, by [39] (*Call for Transparency of COVID-19 Models*, appeared on Science). The challenge arises, therefore, of devising a symbolic model for the diagnosis of COVID-19 based on the acoustic characteristics of a cough/breath sample, whose performances are at least comparable with those of non-symbolic ones.

Structure of the paper. This paper is structured as follows. In Section 2, we briefly review the concept of learning from temporal data. Then, in Section 3, we lay down the foundations of interval temporal decision trees and random forests. Finally, in Section 4, we frame the problem as a binary classification one (that is, the model outputs whether a subject/sample is *positive* or *negative* to COVID-19, similarly to Table 1), solve it with temporal decision trees and random forests, and interpret the resulting models, before concluding.

2. Learning from time series

Time series. 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 numerical or categorical values described as follows:

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

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 datasets are usually numerical. A *temporal dataset* 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 having N points. A *categorical labeled* temporal dataset is a temporal dataset where the instances are associated to a *target variable* $C = \{C_1, \ldots, C_l\}$, also known as *class variable*. In this article, we assume that temporal datasets have no missing values, or that missing values are simply substituted by placeholders. Implicitly, we are also assuming that temporal attributes are all sampled at the same granularity.

Learning from temporal data. As any other learning problem, classification of time series can be approached with functional or symbolic methods. We recall that *functional learning* is the process of learning a *function* that describes the theory of the underlying problem, while *symbolic learning* aims at learning a *logical description* of the same theory. *Ensembles* of classification methods are also common in the learning realm, and while they are usually based on symbolic methods, they may actually be combinations of any learning methods, and it may be argued that they constitute a separate category. As an example, Bagnall et al. [40] developed an ensemble of 35 classifiers for time series classification that includes, among others, *Naive Bayes*, *C4.5 decision trees*, *Support Vector Machines (SVMs)* with linear and quadratic basis function kernels and *Random Forest*, called *Collective Of Transformation-based Ensembles* (*COTE*). Unlike classical (atemporal) learning, classification of time series can also be *instance-based* (i.e., based on the notion of distance/similarity between series) or not, its underlying ontology can be *point-based* or *interval-based*, and the method itself can be *feature-based* (i.e., based on the notion of extracting features from the series, and then, using some atemporal classifier) or not. Finally, a time series classification method may or may not require or allow a *transformation* of the raw data. The plethora of existing methods cannot be immediately partitioned into a taxonomy because *(i)* many proposals present different combinations of these characteristics, and *(ii)* there may be other dimensions that are not properly captured by the above summary. Recently, Ruiz et al. [41] have presented a review and experimental setup for recent algorithmic advances for multivariate time series classification.

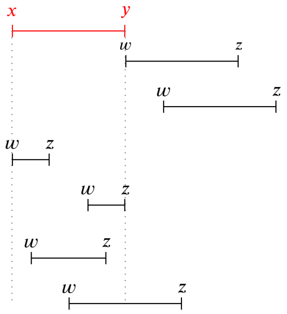
Functional learning. Several function-based approaches have been proposed in the literature. Kakizawa et al. [42] developed optimal bivariate discriminants using multivariate time-invariant forms of discriminant functions. Kudo, in [43], proposed a methodology for classifying sets of data points in a multidimensional space based on the common regions through which only time series of one class pass. Caiado et al. [44] presented a new measure of distance between time series based on the normalized periodogram which estimates the spectral density of a signal. Fulcher and Jones [45] presented a highly-comparative method for learning feature-based classifiers for monovariate time series; their method automatically computes more than 9000 features which are further automatically selected for classification tasks, and the trained model is a linear discriminant classifier that fits a multivariate normal density to each class using a pooled estimate of covariance. The transformation method presented by Moskovitch and Shahar [46] is tested in the same paper using Naive Bayes and Random Forest classifiers, which, to some extent, can be seen as functional-based learning. Functional methods for time series classification in which the notion of distance plays a central role have been developed and tested by Lines and Bagnall [47], in which the classifier is a *Nearest Neighbor* [48,49]. These methods have been also tested on several datasets by Ruiz et al. [41]. A *generative* deep learning model (i.e., an unsupervised model that finds a good representation of the raw time series prior to training a classifier) for time series classification has been proposed by Malhotra et al. [50], where a *Sequence Auto-Encoder (SAE)* based on a sequence-to-sequence model [51] is trained. Wang et al. [52] proposed a simple but strong baseline based on convolutional neural networks (CNNs), with three *discriminative* deep learning models (i.e., models that directly learn the mapping between the raw input time series and the output): multi-layer perceptrons, fully connected networks, and residual networks. The spectrum of deep learning approaches for classifying time series is wide and it is currently a very hot topic in the time series mining research community. A systematic treatment of deep learning methods is beyond the scope of this paper, but an up-do-date and comprehensive review on this topic can be found in Fawaz et al. [53], where the taxonomy for neural networks-based methods from Längkvist et al. [54] is extended.

Symbolic learning. Diez et al. [55] developed an interpretable method for building an ensemble of (base) classifiers with *boosting* [56]. The method extracts a set of rules having only one antecedent. Moreover, point-based and interval-based predicates are defined to cope with the temporal component. In particular, point-based predicates are introduced to test the results obtained with boosting without using interval-based predicates. To some extent, predicates can be seen as features, and this method somehow falls into the realm of *Inductive Logic Programming* (*ILP*). Geurts [57] proposed a feature-based approach that integrates extracted temporal patterns into decision trees. Yamada et al. [58] presented a decision tree-based procedure to classify time series data where the splitting step is done by exhaustively searching a time sequence that is present in data based on class and shape information using *Dynamic Time Warping (DTW)* [59] as dissimilarity measure. A similar approach to Yamada et al.'s is the one proposed by Balakrishan [60] extending regression trees to deal with functional variables (e.g., multivariate time series) and with standard (i.e., non-functional) variables. To split the dataset, representative curves are learned using clustering techniques with similarity measures (i.e., Euclidean Distance and DTW), where the cluster representative is set to be the instance which is closest (i.e., has the smallest combined distance) to all other instances in the cluster, and then, reassign instances to the clusters based on their distance to the representatives (i.e., complete-link hierarchical clustering). Given two sets of signals/time traces (i.e., the good and the bad set), the method proposed by Bartocci et al. [61] finds a logic formula such that it is satisfied with high probability by the good set and with low probability by the bad one. In Baydogan and Runger's [62] work, each (multivariate) time series is represented includes also the first differences (representing trends) for each numerical variable and each row is an instance. Bombara et al. [63] proposed a decision tree-based framework for solving the two-class classification problem involving finite signals (i.e., finite time series) using Signal Temporal Logic classifiers. *Shapelets* [64] have been extensively used in the field of learning from time series; this concept has been used in decision trees to classify time series by Brunello et al. [65]. Finally, Brunello et al. [1] developed a native, interval-based decision tree learner where the instances are represented as timelines.

3. Interval temporal decision trees and forests

Interval temporal logic. While several different interval temporal logics have been proposed in the recent literature (see, e.g., [66]), *Halpern and Shoham's (HS)* [67] is certainly the formalism that has received the most attention, being a very natural choice to model temporal intervals. Although from a logical point of view HS and its fragments have been studied on the most important classes of linearly ordered sets, machine learning datasets are finite structures; therefore,

Table 2
Allen's interval relations and their representation.

HS modality	Definition w.r.t. the interval structure			Example
				 x y w z w z w z w z w z w z
$\langle A\rangle$ (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)	$[x,y]R_O[w,z]$	$\Leftrightarrow$	$x<w<y<z$	

we focus our attention on finite domains. Let $[N]$ be a finite, initial subset of $\mathbb{N}_+$ of cardinality $N>1$, that is, $[N]=\{1,2,\dots,N\}$. 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* [68]: 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 Table 2, 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) [67] 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., $\psi_1\wedge\psi_2\equiv\neg(\neg\psi_1\vee\neg\psi_2)$, $\psi_1\rightarrow\psi_2\equiv\neg\psi_1\vee\psi_2$ 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*) $T=\langle\mathbb{I}([N]),V\rangle$,[1] where $[N]=\{1,2,\dots,N\}$ is a finite linear order, $\mathbb{I}([N])$ is the set of all *(strict) intervals* over $[N]$ with cardinality $N\cdot(N-1)/2$, and V is a *valuation function* $V:\mathcal{AP}\rightarrow 2^{\mathbb{I}([N])}$ 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:

$$\begin{array}{lll} T,[x,y]\Vdash p & \text{iff} & [x,y]\in V(p),\ \text{for each } p\in\mathcal{AP};\\ T,[x,y]\Vdash\neg\psi & \text{iff} & T,[x,y]\nVdash\psi;\\ T,[x,y]\Vdash\psi_1\vee\psi_2 & \text{iff} & T,[x,y]\Vdash\psi_1 \text{ or } T,[x,y]\Vdash\psi_2;\\ T,[x,y]\Vdash\langle X\rangle\psi & \text{iff} & \text{there is } [w,z] \text{ s.t. } [x,y]R_X[w,z] \text{ and } T,\\ & & \quad [w,z]\Vdash\psi; \end{array}$$

where $X\in\mathcal{X}$. Note that, given that the set of relations is jointly exhaustive with respect of $[N]$ (that is, at least one relation holds for each pairs of intervals), the global existential operator, that allows to express global patterns (i.e, "there exists an interval in $[N]$ where ..."), can be defined by the following disjunction:

$$\langle G\rangle\,\varphi := \varphi\vee\bigvee_{X\in\mathcal{X}}\langle X\rangle\,\varphi.$$

Given a model $T=\langle\mathbb{I}([N]),V\rangle$ and a formula φ, we say that T *satisfies* φ if there exists an interval $[x,y]\in\mathbb{I}([N])$ 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*. Observe that HS is interpreted, here, with a *purely interval* semantics, in which the truth of a propositional letter over a given interval does not influence the truth of the same propositional letter over any other interval, regardless of their relative relationship. This is the most general choice, over which we build the theory of interval temporal decision trees; other choices are possible (see, e.g. [69,70]), but are not explored here.

Temporal decision trees. Let $\mathcal{T}=\{T_1,\dots,T_m\}$ be a temporal dataset of m instances, where each is a multivariate time series described by n attributes $\{A_1,\dots,A_n\}$. Given an instance $T\in\mathcal{T}$, we denote the value of an attribute A at the time point t by $A(t)$. Now, let f be a *dynamic feature* function of a variable A; in its simplest form, f is a *scalar descriptor* for A within any interval of the series (e.g., the average value of A over the interval). The key idea of interval temporal decision trees [2] is that decisions are taken over intervals. A temporal decision tree starts off by considering the whole set of instances from an initial time point (e.g., the first temporal value), and computes a predetermined set of dynamic features for each one of the $N\cdot(N-1)/2$ non-point intervals of each series; then, it searches through all possible interval-interval relations, and it establishes which other interval, and which other dynamic feature over that interval, is the most informative in the considered sub-dataset. In this way, it applies the same abstract approach of the classical static decision tree up until a dataset is small enough, or pure enough, so that a stopping criterion can be applied and a leaf can be created. For each possible feature f, let $dom(f(A))$ denote the set of possible values that f takes over A throughout $\mathcal{T}$. The temporal dataset $\mathcal{T}$ entails a propositional alphabet $\mathcal{AP}$ defined as follows:

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

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., in a static dataset, the propositional letter *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 emerges from the continuous nature of the processes described by time series, in which evaluations based on punctual values make little sense. To fix ideas, consider the following example. Given a time interval $[x,y]$ and an attribute A, we could ask whether $A>a$ holds on the interval, which is positively answered if every value of A is higher than a within the interval. However, in order to take full advantage of interval-based semantics, we ask questions that are more generally in the form of $f(A)>a$; this approach can also express questions such as $A>a$ (that is, when f is the

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

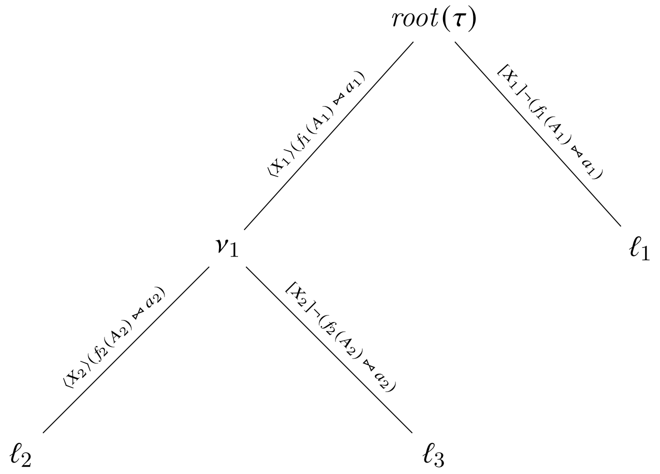

$root(\tau)$
$\langle X_1\rangle(f_1(A_1) \bowtie a_1)$
$[X_1]\neg(f_1(A_1) \bowtie a_1)$
v_1
ℓ_1
$\langle X_2\rangle(f_2(A_2) \bowtie a_2)$
$[X_2]\neg(f_2(A_2) \bowtie a_2)$
ℓ_2
ℓ_3

Fig. 1. Example of a generic temporal decision tree.

minimum function), but it additionally allows the tree to extract more specific information from the interval. As it turns out, the literature on temporal feature extraction provides many scalar functions that can be used as dynamic features (note that any interval of a series is, itself, a series). Therefore, in general, the portfolio of possible feature extraction functions is very wide, and ranges from the simple *minimum*, *maximum*, *average* functions, to more complex statistical functions such as those studied in [71]. In cases where a temporal dataset presents several attributes, choosing which feature functions should be applied to which attribute is a *feature selection* problem, which can be approached, for example, via systematic search, via standard selection techniques (possibly adapted to the temporal case), or simply via an arbitrary choice driven by inductive bias. In temporal decision trees the *univariate split-decisions* (or, simply, *decisions*) that partition a set of instances at a specific node are of type:

$$\mathcal{S} = \{\langle X\rangle(f(A) \bowtie a) \mid X \in \mathcal{X} \cup \{=\}\}.$$

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

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

where $S \in \mathcal{S}$ is a 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 decisions; an object of type $\langle X\rangle(f(A) \bowtie a)$ or of type $\neg\langle X\rangle(f(A) \bowtie a)$ (or, equivalently, $[X]\neg(f(A) \bowtie a)$) is called *edge label*. Decisions (and labels) of type $\langle=\rangle(f(A) \bowtie a)$ are called *propositional*, and they are the standard ones in propositional decision trees. We denote the *root* of τ by $root(\tau)$, and we use $\ell_1, \ell_2, \ldots$ (resp., $v_1, v_2, \ldots$) to denote the *leaves* (resp., *nodes*, both leaf and non-leaf ones). Each non-leaf node v of τ has a *left* (resp., *right*) *child* $L(v)$ (resp., $R(v)$) whose edge is decorated with $S \in \mathcal{S}$, each non-root node v has a *parent* $P(v)$, and each leaf ℓ is labeled with a class, denoted by $C(\ell)$. By convention, we assume that edge labels of type $\langle X\rangle p$ are always *left* (that is, they always label an edge from a node to its left child). A *path* of *length* h between two nodes of τ is a finite sequence of nodes $v_h, v_{h-1}, \ldots, v_0$ such that $v_{i+1} = P(v_i)$, for each $i = 0, \ldots, h-1$; if v_h is $root(\tau)$ and v_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 $v_h \overset{S_h}{\rightsquigarrow} v_{h-1} \overset{S_{h-1}}{\rightsquigarrow} \cdots \overset{S_1}{\rightsquigarrow} v_0$, where $S_i \in \mathcal{S}$, for each $i = 1, \ldots, h$ (see Fig. 1).

Interpreting temporal decision trees. 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 dataset splitting. A *temporal path-formula* $\varphi_{v_h \rightsquigarrow v_0}$ of a path $v_h \overset{S_h}{\rightsquigarrow} v_{h-1} \overset{S_{h-1}}{\rightsquigarrow} \cdots \overset{S_1}{\rightsquigarrow} v_0$, where $S_i \in \mathcal{S}$, in a temporal decision tree τ, is inductively defined on h:

- if $h = 0$, then $\varphi_{v_0 \rightsquigarrow v_0} = \top$;
- if $h > 0$, then let $\varphi_{v_{h-1} \rightsquigarrow v_0} = \xi'_{h-1} \wedge \ldots \wedge \xi'_1 \wedge \top$, and let us call ξ'_i *positive* if it has the form $\xi'_i = \langle X\rangle(f(A) \bowtie a \wedge \psi_i)$, $\xi'_i = (\langle X\rangle(f(A) \bowtie a) \wedge \psi_i)$, or $\xi'_i = (f(A) \bowtie a \wedge \psi_i)$ with $X \in \mathcal{X}$, and *negative* otherwise. Then $\varphi_{v_h \rightsquigarrow v_0}$ is defined by cases:
    - if $v_{h-1} = left(v_h)$, then $\varphi_{v_h \rightsquigarrow v_0} = S_h \wedge \xi_{h-1} \wedge \ldots \wedge \xi_1 \wedge \top$, where, for $1 \leq i \leq h-1$:
        * $\xi_i = \langle X\rangle(f(A) \bowtie a \wedge \xi'_i)$, if $S_h = \langle X\rangle(f(A) \bowtie a)$ and ξ'_i is positive;
        * $\xi_i = (f(A) \bowtie a \wedge \xi'_i)$, if $S_h = f(A) \bowtie a$ and ξ'_i is positive;
        * $\xi_i = (\langle X\rangle(f(A) \bowtie a) \wedge [X](f(A) \bowtie a \rightarrow \xi'_i))$, if $S_h = \langle X\rangle(f(A) \bowtie a)$ and ξ'_i is negative;
        * $\xi_i = (f(A) \bowtie a \wedge (f(A) \bowtie a \rightarrow \xi'_i))$, if $S_h = f(A) \bowtie a$ and ξ'_i is negative;
    - if $v_{h-1} = right(v_h)$, then $\varphi_{v_h \rightsquigarrow v_0} = (S_h) \wedge \xi_{h-1} \wedge \ldots \wedge \xi_1 \wedge \top$, 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 case of static decision trees, from a dataset associated to a node one immediately computes the two datasets that are entailed by a propositional decision. 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 $\mathbb{I}([N]) \cup [0,1]$, denoted by $T.refs$. At the beginning of the learning phase, $T.refs = \{[0,1]\}$ for every T, where $[0,1]$ is an additional, virtual interval 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 potentially entails new reference intervals for every instance of a dataset. 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:

$$\{[w,z] \in \mathbb{I}([N]) \mid \exists [x,y] \in T.refs \wedge [x,y] R_X [w,z] \wedge T, [w,z] \Vdash f(A) \bowtie a\}$$

if $S = \langle X\rangle f(A) \bowtie a$, and as:

$$\{[w,z] \in T.refs \mid T, [w,z] \Vdash f(A) \bowtie a\}$$

if $S = f(A) \bowtie a$. When S is clear from the context, we use $T.refs'$ to denote $f(T.refs, S)$. For a decision $S \in \mathcal{S}$, 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 $\mathcal{T}^{L(v)}$ (respectively, $\mathcal{T}^{R(v)}$). 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 $S \in \mathcal{S}$. 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 dataset $\mathcal{T}$ via the *dataset semantic* relation $\Vdash_\theta$, which generalizes $\Vdash$ from single instances to datasets. 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 [72] seminal work, based on which in [73] the authors proposed their innovative solution as an alternative to functional regression. The algorithm proposed in [74] is the first implementation of a decision tree for classification, but *CART* [75], *ID3* [76], and *C4.5* [77], are the most well-known. All of these algorithms follow the same recursive schema: they start with a tree composed of a single leaf node, associated with the whole dataset; they check for the existence of a split-decision satisfying certain criteria, and if such a decision exists, it is used for splitting the dataset into two sub-datasets (the set of instances satisfying the decision, and the set of instances not satisfying it), which are then passed to the children; the same routine is called on the child nodes. Because all of these algorithms share similar principles, they can be seen as special cases of the general algorithm for *information-based learning* of *temporal decision trees*, which we refer to as *TDT* (see Algorithm 1). Note that information-based learning is a general, greedy and sub-optimal approach to decision tree induction (optimal decision tree induction is knowingly NP-hard [78]). *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 labeled 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:

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

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 dynamic feature f, a specific attribute A, a threshold value $a \in dom(f(A))$, and the operator $\bowtie$. Let $S(f, A, a, \bowtie)$ be the decision entailed by f, A, a, and $\bowtie$, and let $(\mathcal{T}_e, \mathcal{T}_u)$ be the partition of $\mathcal{T}$ entailed by $S(f, A, a, \bowtie)$ (as defined above). The *splitting information* of $S = S(f, A, a, \bowtie)$ is defined as:

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

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

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

In information-based learning, the best candidate split-decision at each node is selected as the one maximizing the information gain. Note that on a dataset with m instances and n attributes the expected (average) time complexity of the algorithm is given by the recursion $T(m, n) = 2 \cdot T(\frac{m}{2}, n) + c(m, n)$, where $c(m, n)$ is the cost of finding the best split-decision at any node. The local cost heavily impacts on the algorithm complexity, and it is a direct function of the number of relations and propositional letters. In the propositional case (i.e., in classic algorithm such as *CART*, *ID3*, *C4.5*), the local cost at a node with j instances is $\mathcal{O}(j \cdot n)$, thus the overall cost is $\mathcal{O}(n \cdot m \log m)$. In the temporal case, the local cost depends on the number of points N, and the overall cost becomes $\mathcal{O}(N(N-1) \cdot n \cdot m \log m)$. Established open-source implementations of decision trees include the classes *DecisionTreeClassifier* in *Scikit-learn* [79] (a Python algorithm implementing *CART*), *J48* in *WEKA* [80] (a Java algorithm implementing *C4.5*), and the *DecisionTree.jl* package [81] available in the Julia programming environment [82]. In [2], an implementation of *TDT* based on *J48* was presented. A more modern and optimized implementation of the same algorithm is available in the Julia package *ModalDecisionTrees.jl* [83] (a fork of *DecisionTree.jl*). Fig. 2 contains a schematic representation of the process of learning a temporal decision tree from a temporal dataset.

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 [3] 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 $(\mathcal{F}, \upsilon)$, where $\mathcal{F}$ is a collection of k temporal decision trees, that is, $\mathcal{F} = \{\tau_1, \dots, \tau_k\}$, and $\upsilon : C^k \to C$ is a *voting aggregation function* of all the unit votes of each temporal decision tree $\tau \in \mathcal{F}$. Given a temporal random forest $(\mathcal{F} = \{\tau_1, \dots, \tau_k\}, \upsilon)$ 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:

$$\upsilon(\tau_1(T), \dots, \tau_k(T)).$$

For a random forest $(\mathcal{F}, \upsilon)$ and a temporal dataset $\mathcal{T}$, 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 [84], using the so-called *random subspace method*. Breiman's proposal [3], which can be considered the standard approach to random forests, was later introduced in the *R* learning package [85], but random forests are part of a more general approach to combine several classifiers into a single one, known as *bagging*, which is still an ongoing research topic, as proven by very recent contributions such as [86], among others. Julia [82] incorporates a class to generalize trees into forests; we used such a class to create a *temporal random forest (TRF)* learning algorithm. *TRF* is based on a generalized version of *TDT* that allows one to use, at each step, only n^{sub} attributes to find the best split while having, at its disposal, only m^{sub} instances, as shown in Algorithm 1; this is a *randomized* version of the interval temporal logic decision tree learning strategy, which degenerates into the deterministic version when $n^{sub} = n$ and $m^{sub} = m$. 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 *TDT* 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 lookup table keyed on the tuple $(T, [x, y], X, A, a, f)$, that returns the truth value of the decision $\langle X \rangle (f(A) \bowtie a)$ of the instance T on the interval $[x, y]$. 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 *TDT* without costly recomputations, effectively improving the overall experimental complexity with respect to k independent executions of *TDT*. An implementation of temporal decision forests is available in the *ModalDecisionTrees.jl* Julia package.

4. Data and experiments

Dataset. The dataset used in this paper, presented in [8], was originally crowdsourced and compiled by researchers at the University of Cambridge, and it is available upon request; in Table 1 we refer to as CAM. It has the following structure. The dataset in its entirety is composed of 9986 audio samples recorded by 6613 volunteers. Each audio recording 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. Out of all volunteers, 235 declared to be *positive* to COVID-19. The subjects are quasi-normally distributed by

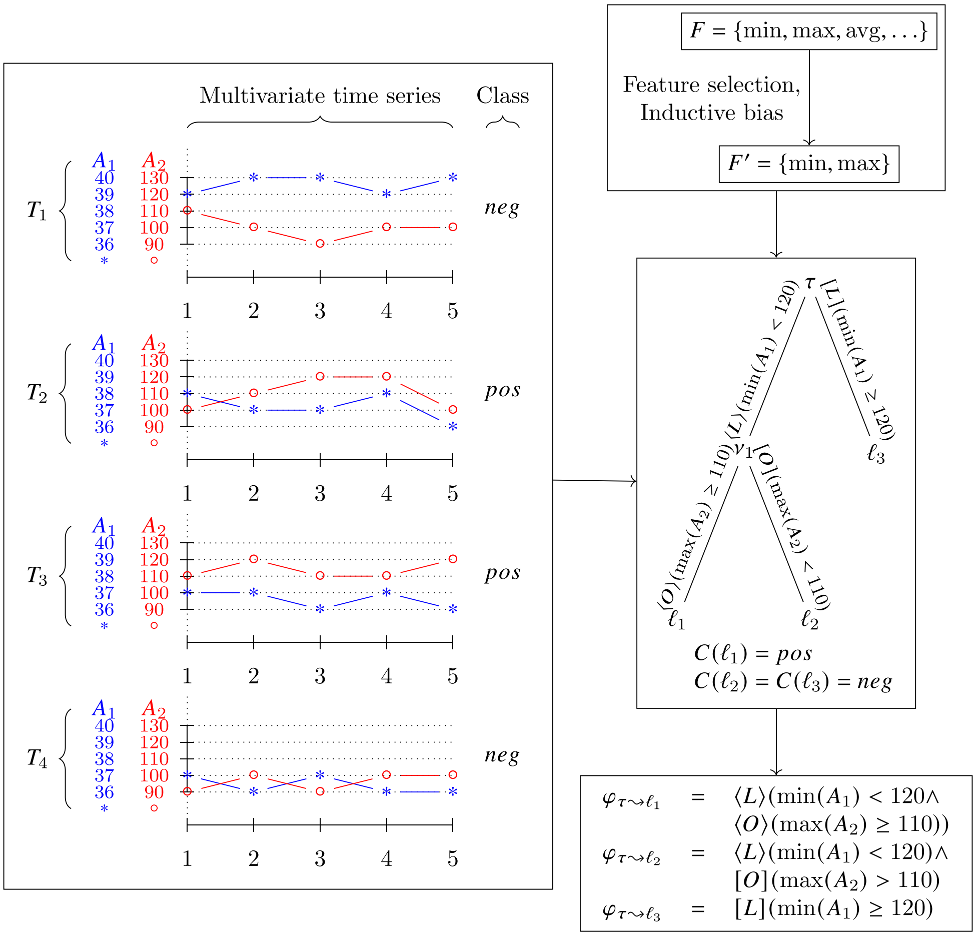

Multivariate time series
Class
T_1
T_2
T_3
T_4
A_1
A_2
40 39 38 37 36
130 120 110 100 90
1 2 3 4 5
neg
pos
pos
neg
$F = \{\min, \max, \text{avg}, \ldots\}$
Feature selection, Inductive bias
$F' = \{\min, \max\}$
τ
$\langle L \rangle(\min(A_1) < 120)$
$[L](\min(A_1) \geq 120)$
ν_1
ℓ_3
$\langle O \rangle(\max(A_2) \geq 110)$
$[O](\max(A_2) < 110)$
ℓ_1
ℓ_2
$C(\ell_1) = pos$
$C(\ell_2) = C(\ell_3) = neg$
$\varphi_{\tau \rightsquigarrow \ell_1} = \langle L \rangle(\min(A_1) < 120 \wedge \langle O \rangle(\max(A_2) \geq 110))$
$\varphi_{\tau \rightsquigarrow \ell_2} = \langle L \rangle(\min(A_1) < 120) \wedge [O](\max(A_2) > 110)$
$\varphi_{\tau \rightsquigarrow \ell_3} = [L](\min(A_1) \geq 120)$

Fig. 2. A schematic representation of the process of learning from a temporal dataset.

Algorithm 1: High-level description of *TDT*.

```
function $TDT(\mathcal{T}, n^{sub})$:
    $\tau \leftarrow$ initialize an empty decision tree
    $Preprocess(\mathcal{T})$
    $root(\tau) \leftarrow Learn(\mathcal{T}, n^{sub})$
    return $\tau$
end
function $Learn(\mathcal{T}, n^{sub})$:
    if a stopping condition applies then
        return $CreateLeafNode(\mathcal{T})$
    end
    $S \leftarrow FindBestDecision(\mathcal{T}, n^{sub})$
    $(\mathcal{T}_e, \mathcal{T}_u) \leftarrow Split(\mathcal{T}, S)$
    $\nu \leftarrow CreateNode(\mathcal{T})$
    $L(\nu) \leftarrow Learn(\mathcal{T}_e)$
    $R(\nu) \leftarrow Learn(\mathcal{T}_u)$
    return $\nu$
end
```

Algorithm 2: High-level description of *TRF*.

```
function $TRF(\mathcal{T}, k, m^{sub}, n^{sub})$:
    $\mathcal{F} \leftarrow \emptyset$
    foreach $i \in [1, \ldots, k]$ do
        $\mathcal{T}' \leftarrow SubsetSample(\mathcal{T}, m^{sub})$
        $\tau \leftarrow TDT(\mathcal{T}', n^{sub})$
        $\mathcal{F} \leftarrow \mathcal{F} \cup \{\tau\}$
    end
    return $\mathcal{F}$
end
```

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. Beside recording sound samples, subjects were asked to fill in a very brief clinical history, plus information about their geographical location. In [8], this data was originally used to derive smaller datasets, each posing a different form of the same task of COVID-19 diagnosis. In particular, the location of the subject was 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 diagnostic test for COVID-19, only a subset of the subjects who declared to be negative could indeed be reliably considered as negative. With this approach, three tasks were designed: *(i)* to distinguish between declared positive subjects and non-positive ones that have a *clean medical history*, have *never smoked*, have *no symptoms*, and live in countries in which the virus spread at that moment was very low (so they can be reliably considered negative subjects); *(ii)* to distinguish between declared positive subjects with *cough as symptom* and non-positive ones that have a *clean medical history*, have *never smoked*, have *cough as a symptom*, and live in countries in which the virus spread at that moment was very low (so they can be reliably considered negative subjects with a cough); *(iii)* to distinguish between declared positive subjects with *cough as symptom* and non-positive ones that have *asthma*, that have *never smoked*, have *cough* *as a symptom*, and live in countries in which the virus spread at that moment was very low (so they can be reliably considered negative subjects with cough and asthma). We refer to these tasks and datasets as $TA1, TA2$, and $TA3$. To counteract the small amount of control data, the authors of the original dataset also produced and released two *augmented* versions for $TA2$ and $TA3$ (referred, here, as $TA2+$ and $TA3+$, respectively), obtained with standard audio augmentation methods. In [8], the authors considered three versions of each task, which differ by how subjects are represented, that is, using only their cough sample, only their breath sample, or both; note that, although in their original release temporal decision trees and forests were not designed for *multi-frame* (also known and *multimodal*) representation of the data, such an extension was studied in [87], thus these models are also able to treat subjects represented as the union of a cough and a breath sample. With respect to the original work, we have eliminated 14 instances that presented cough/breath labeling mistakes, empty audio recording, and/or a too noisy background; barring such difference, it is possible to directly compare our results with the ones from the original paper and from other models trained on the same data. Moreover, because of the nature of interval temporal trees, it also makes sense to explore learning from *single* coughs/breath cycles, instead of whole recordings which present several episodes; for each version of the dataset, we produced a *segmented* variant containing single episodes from the original ones.

Pre-processing techniques. In audio signal processing, it is customary to extract *spectral* representations of sounds, facilitating 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 [88], is still the preferred technique for extracting sensible spectral representations of audio data, and its use in machine learning has been fruitful for tackling difficult 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. 25 ms), 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 converted and represented in the so-called *mel scale*, a logarithmic representation which causes the frequency space to better reflect human ear perception of frequencies; *(iii)* a set of n triangular band-pass filters is convolved across the frequency spectrum, discretizing it into a finite number of frequencies; *(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 audio 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 (e.g., SVMs, neural networks). Moreover, step (iv) does not preserve the spectral component, which makes this description not directly interpretable in terms of sound frequencies; as such, we applied MFCC up to step (iii), ultimately obtaining a multivariate time series representation where the n attributes describe the power of the different sound frequencies. Furthermore, different techniques were used to clean and normalize the data prior to the MFCC step: *(i)* a *noise gate* filter to attenuate signals that register below a threshold to remove background noises; *(ii)* a *peak normalization* filter where the amplitude is scaled on the highest signal level present in the recording granting consistent amplitude between audio tracks; *(iii)* silence removal filter to remove unwanted long silences. Additionally, in order to make the model invariant to different *tones*, a *pitch normalization* step was applied, where instead of the *mel scale*, the frequency spectrum was represented via the *semitone scale*, which is still logarithmic, but relative to a fundamental frequency. Such a fundamental frequency for each sample was found by means of a Fast Fourier Transform (FFT) as the most prevalent frequency between 200 Hz and 700 Hz, which is generally accepted as appropriate for human cough in normal conditions [89,90].

Experimental settings. For each of the 30 problems described in a previous paragraph, a number of temporal decision trees and temporal random forests were trained and evaluated via standard performance metrics for binary classification: overall accuracy (*acc*), precision (*prec*), precision (*rec*) and F1-score ($F1$). To minimize the bias, datasets were balanced by downsampling the majority class, and randomly split into two (balanced) sets for training (80%) and test (20%). This process was repeated 10 times (randomized 10-folds cross-validation), and the average and standard deviation of the performance metrics were considered. Furthermore, for any training/test split, random forests were run 5 times with different initial random conditions, and their average performance was considered. As for the parametrization of random forests, after a pre-screening phase, we set $m^{sub} = 70\%$ of the cardinality of each training set (m), $k = 100$, and $n^{sub} = 50\%$ of the number of attributes (n). While experiments for single decision trees were run with a standard pre-pruning setting, that is, minimum entropy gain of 0.01 for a split to be meaningful and maximum entropy at each leaf of 0.6, random forests grow full trees. In all cases we let $\bowtie$ be in $\{\leq, \geq\}$. As for f, as we have already mentioned, there are many possible choices; in order to maximize the interpretability of our models, however, we used only *minimum* and *maximum*, in their *softened* version, which correspond to the $20th$ and the $80th$ percentile, respectively; thus $f \in \{\min_{80}, \max_{80}\}$. As for the pre-processing, the chunk size and overlap for the DFT were fixed to the standard values of 25 ms and 10 ms, respectively. Pre-screening was also applied to the parameters that partially drive the form of the data, that is, number of frequencies (i.e., the number of attributes n), the size of the moving average filter (w) used to lower the number of resulting points, and step of the moving window (s); as a result of such a pre-screening, we fixed $n = 30$, $w = 30$, and $s = 20$. In any case, for further data dimensionality reduction, the resulting series were capped at a maximum of 50 time points each. The pre-screening also found that noise gate, peak normalization and silence removal were effective for cough samples, while peak normalization and silence removal were effective for breath samples. Pitch normalization proved to be effective both with cough and breath samples; furthermore, all audio samples with a sample rate lower than 16 000 Hz were discarded. Experiments were run in a Julia environment, using open-source packages: more specifically, the *WAV.jl*, *WORLD.jl*, *DSP.jl*, and *ImageFiltering.jl* packages for audio signal processing, and the *ModalDecisionTrees.jl* package [83] for training temporal trees and forests.

Results at a glance. The following questions are interesting: Are temporal decision trees and temporal random forest suitable methods to solve this problem? Which combination of parameters gives the best results? Are our best results comparable with the results obtained by standard techniques, as presented in [8] (the original work on the CAM dataset) and in more recent works on the same dataset (see Table 5)? How do our results compare with others in terms of tradeoff between the complexity of the training/model and the performances? How can our results be interpreted? As far as the suitability of our method is concerned, let us focus on Tables 3 and 4. As already mentioned, each row is the average of 10 executions of a specific combination of dataset settings; each performance is associated with its experimental standard deviation, for a better assessment of the solidity of the results. It is immediately clear that the datasets with segmented coughs and breath cycles perform better than the original ones. This is probably due to two aspects: first, temporal decision trees and random forests can focus on the relevant acoustic aspects of positive versus negative samples with a single episode at the time; second, segmented datasets are generally much bigger than non-segmented ones, which allowed us to train better models. We have therefore followed two different rules to highlight the results in Tables 3 and 4: for the non-segmented datasets, rows with accuracies better than 85% have been highlighted,

Table 3
Results obtained using original data (non-segmented).

			acc	*prec*	*rec*	$F1$	m_{train}	m_{test}
Decision tree	Cough	$TA1$	67.8 ± 8	68.9 ± 9	66.0 ± 9	67.2 ± 8	202	50
		$TA2$	79.0 ± 14	81.4 ± 18	80.0 ± 13	79.6 ± 12	40	10
		$TA3$	60.0 ± 21	65.0 ± 25	70.0 ± 26	63.7 ± 17	14	4
		$TA2+$	83.3 ± 6	82.1 ± 8	86.7 ± 11	83.7 ± 6	74	18
		$TA3+$	**90.6 ± 6**	96.4 ± 6	84.5 ± 11	89.7 ± 7	72	18
	Breath	$TA1$	68.8 ± 6	70.7 ± 7	65.2 ± 10	67.4 ± 7	202	50
		$TA2$	82.0 ± 11	87.8 ± 14	76.0 ± 16	80.5 ± 12	36	10
		$TA3$	55.0 ± 16	53.7 ± 26	60.0 ± 39	51.0 ± 29	12	4
		$TA2+$	82.8 ± 9	86.1 ± 16	83.3 ± 12	83.2 ± 8	74	18
		$TA3+$	**85.0 ± 9**	93.5 ± 9	75.0 ± 16	82.6 ± 12	64	16
	Cough+breath	$TA1$	66.4 ± 7	67.4 ± 8	64.4 ± 6	65.8 ± 6	202	50
		$TA2$	77.0 ± 15	81.5 ± 19	74.0 ± 16	76.4 ± 14	36	10
		$TA3$	65.0 ± 13	70.0 ± 22	75.0 ± 26	67.3 ± 11	12	4
		$TA2+$	**85.0 ± 9**	86.1 ± 10	84.5 ± 9	85.0 ± 9	74	18
		$TA3+$	84.4 ± 4	91.7 ± 7	76.3 ± 9	82.8 ± 5	64	16
Random forest	Cough	$TA1$	76.6 ± 7	79.4 ± 9	72.4 ± 8	75.6 ± 8	202	50
		$TA2$	83.4 ± 12	85.3 ± 14	83.6 ± 18	83.2 ± 13	40	10
		$TA3$	70.5 ± 19	79.3 ± 23	70.0 ± 35	66.5 ± 28	14	4
		$TA2+$	**88.7 ± 6**	91.9 ± 7	85.3 ± 10	88.1 ± 7	74	18
		$TA3+$	**89.2 ± 7**	96.3 ± 6	81.6 ± 11	87.9 ± 8	72	18
	Breath	$TA1$	74.5 ± 6	76.3 ± 9	72.3 ± 7	74.0 ± 6	202	50
		$TA2$	84.0 ± 10	88.9 ± 12	80.0 ± 19	82.7 ± 11	36	10
		$TA3$	62.0 ± 21	63.0 ± 32	63.0 ± 33	59.7 ± 27	12	4
		$TA2+$	**87.9 ± 6**	91.9 ± 8	84.0 ± 11	87.2 ± 6	74	18
		$TA3+$	**94.5 ± 4**	98.9 ± 4	90.3 ± 9	94.0 ± 5	64	16
	Cough+breath	$TA1$	74.8 ± 6	76.1 ± 8	73.0 ± 7	74.3 ± 6	202	50
		$TA2$	84.2 ± 10	87.3 ± 10	81.2 ± 18	83.0 ± 11	36	10
		$TA3$	69.5 ± 19	76.0 ± 26	69.0 ± 25	68.6 ± 18	12	4
		$TA2+$	**89.8 ± 5**	93.5 ± 5	85.8 ± 11	89.1 ± 6	74	18
		$TA3+$	**89.5 ± 7**	95.4 ± 6	83.3 ± 11	88.5 ± 8	64	16

Table 4
Results obtained using segmented data.

			acc	*prec*	*rec*	$F1$	m_{train}	m_{test}
Decision tree	Cough	$TA1$	72.8 ± 8	75.0 ± 8	68.4 ± 9	71.4 ± 8	340	86
		$TA2$	91.0 ± 10	96.7 ± 11	86.0 ± 14	90.3 ± 10	42	10
		$TA3$	72.5 ± 25	79.6 ± 26	70.0 ± 35	68.3 ± 32	12	4
		$TA2+$	93.2 ± 8	92.3 ± 8	94.6 ± 12	93.1 ± 9	92	22
		$TA3+$	90.0 ± 5	92.4 ± 7	87.5 ± 8	89.7 ± 6	64	16
	Breath	$TA1$	74.3 ± 3	74.6 ± 4	74.1 ± 3	74.3 ± 2	982	246
		$TA2$	84.0 ± 5	84.0 ± 6	84.7 ± 11	83.9 ± 6	120	30
		$TA3$	61.7 ± 19	61.0 ± 30	66.7 ± 35	59.9 ± 27	20	6
		$TA2+$	88.2 ± 4	91.3 ± 4	84.6 ± 9	87.5 ± 5	326	82
		$TA3+$	91.9 ± 7	98.4 ± 3	85.4 ± 13	90.8 ± 9	104	26
	Cough+breath	$TA1$	**95.9 ± 1**	96.5 ± 2	95.3 ± 2	95.9 ± 2	1338	334
		$TA2$	**98.8 ± 2**	100.0 ± 0	97.7 ± 4	98.8 ± 2	102	26
		$TA3$	90.0 ± 17	92.6 ± 15	90.0 ± 32	86.0 ± 31	12	4
		$TA2+$	**97.8 ± 2**	98.5 ± 1	97.0 ± 4	97.7 ± 2	430	108
		$TA3+$	84.4 ± 18	87.1 ± 21	86.3 ± 15	85.6 ± 16	64	16
Random forest	Cough	$TA1$	80.4 ± 6	84.0 ± 6	75.2 ± 8	79.2 ± 6	340	86
		$TA2$	92.4 ± 7	99.3 ± 2	85.6 ± 13	91.4 ± 8	42	10
		$TA3$	73.5 ± 23	78.0 ± 25	77.0 ± 24	74.6 ± 20	12	4
		$TA2+$	**95.5 ± 4**	98.5 ± 3	92.6 ± 9	95.1 ± 5	92	22
		$TA3+$	92.9 ± 7	100.0 ± 0	85.8 ± 15	91.5 ± 10	64	16
	Breath	$TA1$	81.9 ± 2	84.0 ± 3	79.0 ± 3	81.4 ± 2	982	246
		$TA2$	86.7 ± 7	91.5 ± 7	82.3 ± 16	85.4 ± 9	120	30
		$TA3$	66.3 ± 12	68.1 ± 19	67.3 ± 31	63.7 ± 19	20	6
		$TA2+$	90.5 ± 3	95.7 ± 3	84.9 ± 6	89.9 ± 3	326	82
		$TA3+$	92.0 ± 8	99.9 ± 0	84.2 ± 16	90.5 ± 11	104	26
	Cough+breath	$TA1$	**98.0 ± 0**	99.4 ± 1	96.7 ± 1	98.0 ± 0	1338	334
		$TA2$	**97.2 ± 3**	100.0 ± 0	94.5 ± 6	97.0 ± 3	102	26
		$TA3$	86.5 ± 16	93.7 ± 9	81.0 ± 32	81.5 ± 28	12	4
		$TA2+$	**99.4 ± 1**	99.0 ± 1	99.9 ± 0	99.4 ± 1	430	108
		$TA3+$	**96.1 ± 5**	100.0 ± 0	92.3 ± 11	95.6 ± 6	64	16

while for the segmented ones, we have focused our attention on rows having accuracy higher than 95%. A second, immediate observation is that multi-frame learning performs decidedly better than standard, single-audio learning; this means that positive samples are more easily recognized from negative ones from a combination of (a single) breath and cough episode than they are from breath and cough separately; the performances of the models on the tasks $TA1$ and $TA2$, in particular, benefit from this approach. This is consistent with the results obtained

Table 5
Performances of classification models on CAM: state of the art.

	acc	*prec*	*rec*	*F1*	*AUC*	*acc*	*prec*	*rec*	*F1*	*AUC*
	Brown et al. [8]					[33]'s implementation of [27]				
TA1	–	72	69	–	80	60	54	59	54	56
TA2	–	**80**	**72**	–	**82**	–	–	–	–	–
TA3	–	**69**	**69**	–	**80**	–	–	–	–	–
TA2+	–	70	90	–	87	72	82	85	90	66
TA3+	–	61	81	–	88	86	82	86	83	88
	Coppock et al. [27]					[33]'s implementation of [19]				
TA1	–	–	–	–	**83**	72	71	72	71	68
TA2	–	–	–	–	–	–	–	–	–	–
TA3	–	–	–	–	–	–	–	–	–	–
TA2+	–	–	–	–	68	88	86	88	84	66
TA3+	–	–	–	–	91	90	90	90	88	76
	Dentamaro et al. [33]					[33]'s implementation of [16]				
TA1	**80**	**80**	**80**	80	**83**	71	71	71	68	78
TA2	–	–	–	–	–	–	–	–	–	–
TA3	–	–	–	–	–	–	–	–	–	–
TA2+	**93**	**89**	**93**	90	**93**	85	82	85	90	88
TA3+	**93**	**89**	**93**	90	**93**	85	78	85	81	87

in [8]. As a third consideration, we notice, as expected, that temporal random forests perform consistently better than their single tree counterpart; yet, very high accuracies are obtained with single trees in some cases. In accordance with [8], augmented datasets give rise to better models in virtually all cases. The worst results emerge from $TA3$ (arguably, the most challenging among the three problems), partly because of the intrinsic difficulty of the problem, but most importantly because of the small size of datasets, which effects are worsened by the downsampling step; $TA3+$, its augmented counterpart, however, allowed us to train much better models. The best result with non-segmented datasets and single trees has been obtained precisely with $TA3+$, with an average accuracy of 90.6%; in this case, temporal random forests do not improve the accuracy (best accuracy in this case: 89.2%). As for segmented datasets, the best result with single trees is a very notable 98.8%, obtained in $TA2$, with only 2% of standard deviation over 10 executions; this result is already unmatched in the current literature (see Table 5). With temporal random forests, $TA2+$ allowed us to obtain an astonishing 99.4% of averaged accuracy, 1% of standard deviation, which is the best known performance of a COVID-19 acoustic diagnostic system, across all existing datasets and learning methods (see Table 1); in the segmented, multi-frame setting, however, temporal random forest models have accuracies better than 96% for all tasks, except for $TA3$, which is an indication of the reliability of this method. The multi-frame approach, as it can be seen, also has the effect of lowering the standard deviation across the different sets of experiments, at least in the segmented case. In order to evaluate which of the sample types (cough, breath, or cough+breath) is most suited for this kind of prediction, and whether multi-frame learning is statistically more effective than standard learning, the accuracy results for the different versions can be compared using the *Friedman test* [91]. In addition to this, a *Wilcoxon signed-rank test* [92] can be performed on each pair of versions to check if there are significative differences between their results. The results of such tests, corrected using the *Holm–Bonferroni method* [93], are shown in terms of *critical difference* [94,95] diagrams in Fig. 3. It is worth noting that in the segmented case, although the multi-frame approach does not always show a critical difference with respect to the single-sample cases, it shows a lower *average rank* in almost all tasks for the segmented dataset (that is, in all tasks except for $TA3+$ using the decision tree), indicating a better performing model. A similar trend is highlighted in Fig. 4, where the critical differences between decision trees and random forests across the two segmented and non-segmented versions are shown. All the experiments were executed on a computing server with *Intel® Xeon® Gold 6238R* using 32 threads, and the average time required to train the models for each task can be seen in Table 6. Note that unrelated experiments were simultaneously run on the same server, and the reported times may be affected by this.

Performances and interpretation of single trees. Besides our ability to distinguish between positive and negative cases, it is important to discuss the structure of the models themselves. The current trend is to solve every learning problem using modern functional approaches (as we have already noticed, in the case of acoustic-based diagnosis models the functional approach is at the core of virtually all recent contributions). More specifically, in the majority of cases acoustic functional models are based on (deep) neural networks, and make use of very complex acoustic features, often extracted with pre-trained models; this combination of functional approach and complex acoustic features produces statistically reliable, but essentially non-interpretable models, which in most cases are also computationally very demanding. Recall that we have extracted models from 30 variants of the original CAM dataset, with each extraction consisting of 10 models, which means that, in terms of single decision trees, we have a pool of 300 trained models. Among them, and in particular from those obtained for $TA2+$ in the segmented case (in which case we have reached, as already observed, an accuracy of 99.4% on average), we have selected a single tree with 100% test accuracy; it is shown in Fig. 5. As it can be seen, such a tree is in fact a very simple model, in which most of the examples fall in one of two leaves. These induce two rules, one for positive cases (pos), with a support of 42%, and one negative ones (neg), with a support of 48%, as follows:

$$
\begin{array}{lll}
R1 & \{b\}\langle G\rangle(\min_{80}(A_4) \geq 1.15) \wedge & \\
 & \{c\}[G]\neg(\min_{80}(A_{15}) \geq 1.35 \times 10^6) & \Rightarrow \; neg \\
 & & \\
R2 & \{b\}[G]\neg(\min_{80}(A_4) \geq 1.15) \wedge & \\
 & \{b\}\langle G\rangle(\min_{80}(A_{14}) \geq 6.81 \times 10^6) \wedge & \\
 & \{b\}\langle G\rangle(\min_{80}(A_{14}) \geq 6.81 \times 10^6 \wedge \langle \overline{L}\rangle(\max_{80}(A_{22}) \leq 7 \times 10^5)) & \Rightarrow \; pos
\end{array}
$$

The above rules are a perfect example of interpretable diagnosis model: as it can be seen, they relate frequencies (in the cough and/or in the breath sample) and powers, and allow one to formally define the *essence* of being COVID-19 positive or negative. In particular, frequencies $A_4 \approx 43$ Hz, $A_{14} \approx 299$ Hz, $A_{15} \approx 363$ Hz, and $A_{22} \approx 1405$ Hz are the ones that have been found particularly relevant for this classification task.

5. Conclusions

The ability of explaining the underlying theory that is extracted with machine learning methods is of uttermost importance, especially in medical applications. Interpretability and explainability in learning are often synonymous of a symbolic approach, which, in turn, should be based on logics that are able to grasp the complexity of the phenomena.

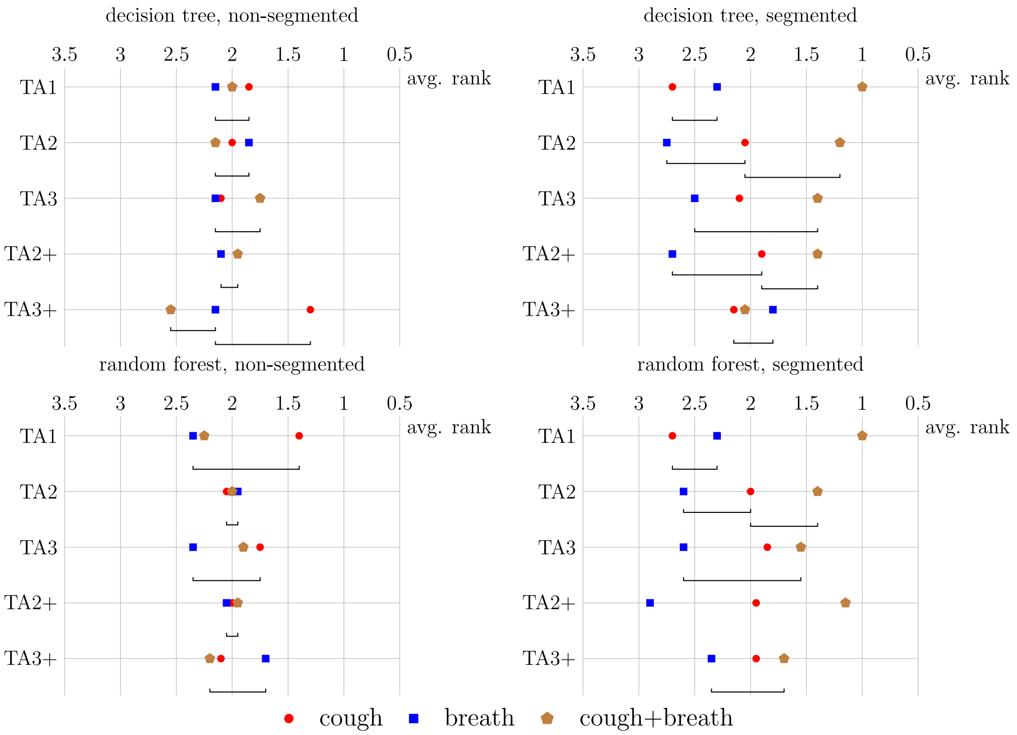


Fig. 3. Critical difference diagrams comparing accuracies of the three sample types (cough, breath and cough+breath) for each task.

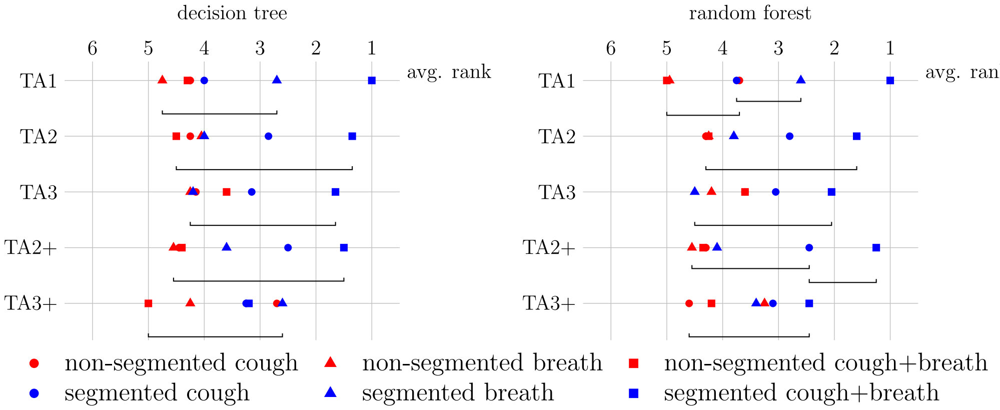


Fig. 4. Critical difference diagrams comparing accuracies of the segmented and non-segmented versions for each task.

Modal symbolic learning offers classical learning tools enhanced with modal propositional logics that allow one to capture complex patterns from data; temporal symbolic learning is the specialization of modal symbolic learning to the case of temporal data and temporal logics. In this paper, we used temporal decision trees and temporal random forests, which are based on Halpern and Shoham's interval temporal logic HS, to build models for the diagnosis of COVID-19 based on the acoustic characteristics of cough/breath samples of positive and negative subjects, interpreted as a multivariate time series. We found that not only is our approach completely innovative, but its performances are superior to those of classical methodologies, both symbolic and functional, applied to the same data, while allowing for the interpretation of the results and enabling visualization (and transformation into audible sounds) of the models that encloses 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 screening, for example as a smartphone application. As a future work, the next natural step would be how our approach generalizes to other datasets, not only for the case of COVID-19, but also for the case of other respiratory diseases.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research has been partially founded by the FIRD, Italy project *Modal Geometric Symbolic Learning*, founded by the University of Ferrara and the GNCS, Italy project *Natural Language Elaboration and Temporal Logic for Text Formalization*, founded by INDAM, Italy. Moreover, the authors would like to acknowledge the help of Dr. Muhammad Azfar Imtiaz for his help during the reaction of the final version of this paper.

Table 6
Average computational time required to train each model. For each experiment, the number of training instances and their length (i.e., number of points) is also shown. Note that, while the average time for decision tree models is computed over the 10 cross-validation repetitions, each random forest is computed 5 times with different random seeds, thus the average time is computed over a total of 50 experiments.

		Non-segmented				Segmented			
		Train time (s)		m_{train}	N	Train time (s)		m_{train}	N
		Decision tree	Random forest			Decision tree	Random forest		
Cough	$TA1$	15.89	1,654.57	202	50	4.22	136.81	340	3
	$TA2$	1.05	53.34	40	37	0.18	2.06	42	3
	$TA3$	0.09	6.28	14	37	0.01	0.15	12	3
	$TA2+$	2.89	247.03	74	37	0.40	5.94	92	3
	$TA3+$	1.07	96.57	72	37	0.22	2.94	64	3
Breath	$TA1$	110.30	5,184.86	202	50	122.47	8,271.27	982	44
	$TA2$	4.28	238.98	36	50	3.54	124.35	120	26
	$TA3$	0.10	80.99	12	50	0.23	5.03	20	26
	$TA2+$	10.91	1,079.55	74	50	13.50	947.30	326	26
	$TA3+$	15.51	598.27	64	50	2.78	101.68	104	26
Cough+breath	$TA1$	56.07	5,355.72	202	50, 50	82.65	8,717.38	1338	3, 44
	$TA2$	1.09	170.95	36	37, 50	1.10	28.56	102	3, 26
	$TA3$	0.08	22.40	12	37, 50	0.01	0.67	12	3, 26
	$TA2+$	3.58	1,048.96	74	37, 50	14.08	905.60	430	3, 26
	$TA3+$	6.11	667.19	64	37, 50	0.14	21.10	64	3, 26

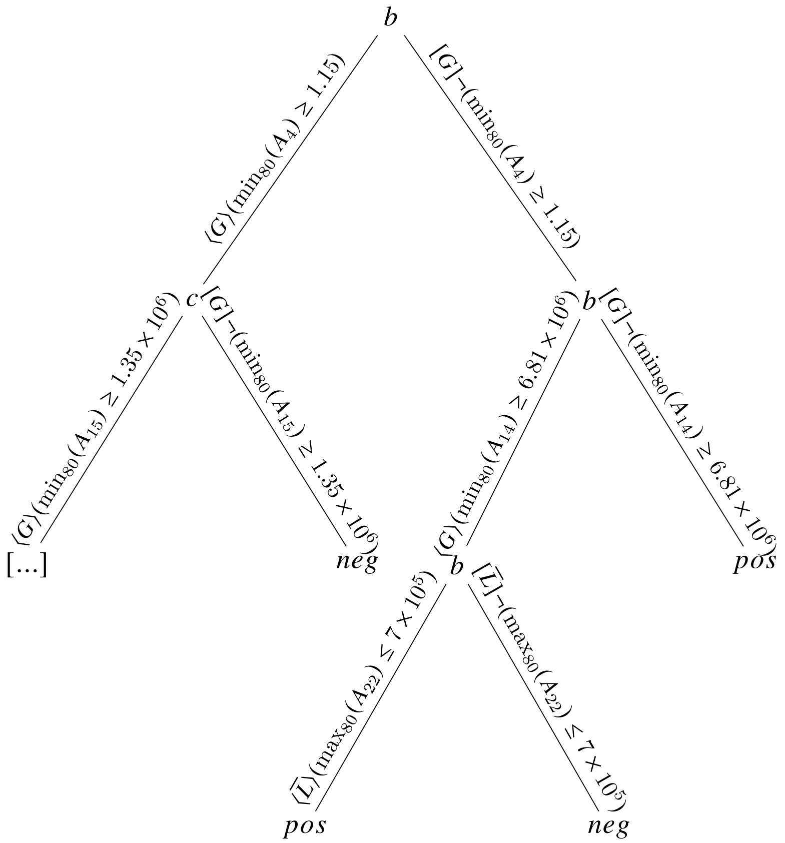

b
$\langle G \rangle(\min_{80}(A_4) \geq 1.15)$
$[G]\neg(\min_{80}(A_4) \geq 1.15)$
c
$\langle G \rangle(\min_{80}(A_{15}) \geq 1.35 \times 10^6)$
$[G]\neg(\min_{80}(A_{15}) \geq 1.35 \times 10^6)$
b
$\langle G \rangle(\min_{80}(A_{14}) \geq 6.81 \times 10^6)$
$[G]\neg(\min_{80}(A_{14}) \geq 6.81 \times 10^6)$
[...]
neg
b
pos
$\langle \overline{L} \rangle(\max_{80}(A_{22}) \leq 7 \times 10^5)$
$[\overline{L}]\neg(\max_{80}(A_{22}) \leq 7 \times 10^5)$
pos
neg

Fig. 5. A temporal decision tree with 100% test accuracy from task $TA2+$ (segmented, cough+breath, 4th fold) using both cough and breath samples.

References

[1] Brunello A, Sciavicco G, Stan IE. Interval temporal logic decision tree learning. In: Proc. of the 16th European conference on logics in artificial intelligence. JELIA, Lecture notes in computer science, vol. 11468, Springer; 2019, p. 778–93.

[2] Sciavicco G, Stan IE. Knowledge extraction with interval temporal logic decision trees. In: Proc. of the 27th international symposium on temporal representation and reasoning. TIME, Leibniz international proceedings in informatics, vol. 178, 2020, p. 9:1–9:16.

[3] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.

[4] Friedman JH, Popescu BE. Predictive learning via rule esambles. Ann Appl Stat 2008;2(3).

[5] Meinshausen N. Node harvest. Ann Appl Stat 2010;4(4).

[6] Deng H. Interpreting tree ensembles with inTrees. Int J Data Sci Anal 2019;7:277–89.

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

[8] Brown C, Chauhan J, Grammenos A, Han J, Hasthanasombat A, Spathis D, Xia T, Cicuta P, Mascolo C. Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data. In: Proc. of the 26th ACM SIGKDD international conference on knowledge discovery and data mining. KDD, 2020, p. 3474–84. http://dx.doi.org/10.1145/3394486.3412865.

[9] Chaudhari G, Jiang X, Fakhry A, Han A, Xiao J, Shen S, Khanzada A. Virufy: Global applicability of crowdsourced and clinical datasets for AI detection of COVID-19 from cough. 2020, CoRR abs/2011.13320. arXiv:2011.13320.

[10] Orlandic L, Teijeiro T, Atienza D. The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms. Sci Data 2021;8.

[11] Sharma N, Krishnan P, Kumar R, Ramoji S, Chetupalli SR, R. N, Ghosh PK, Ganapathy S. Coswara – a database of breathing, cough, and voice sounds for COVID-19 diagnosis. In: Proc. of the 21st annual conference of the international speech communication association (INTERSPEECH). 2020, p. 4811–5.

[12] Cohen-McFarlane M, Goubran R, Knoefel F. Novel Coronavirus Cough Database: NoCoCoDa. IEEE Access 2020;8:154087–94.

[13] Xia T, Spathis D, Brown C, Ch J, Grammenos A, Han J, Hasthanasombat A, Bondareva E, Dang T, Floto A, Cicuta P, Mascolo C. COVID-19 sounds: A large-scale audio dataset for digital COVID-19 detection. In: Proc. of the 35th conference on neural information processing systems (NIPS) datasets and benchmarks track (Round 2). 2021.

[14] Muguli A, Pinto L, R N, Sharma N, Krishnan P, Ghosh PK, Kumar R, Bhat S, Chetupalli SR, Ganapathy S, Ramoji S, Nanda V. DiCOVA challenge: Dataset, task, and Baseline system for COVID-19 diagnosis using acoustics. In: Proc. of the 22nd annual conference of the international speech communication association (INTERSPEECH). 2021, p. 901–5.

[15] Schuller BW, Batliner A, Bergler C, Mascolo C, Han J, Lefter I, Kaya H, Amiriparian S, Baird A, Stappen L, Ottl S, Gerczuk M, Tzirakis P, Brown C, Chauhan J, Grammenos A, Hasthanasombat A, Spathis D, Xia T, Cicuta P, Rothkrantz LJM, Zwerts JA, Treep J, Kaandorp CS. The INTERSPEECH 2021 computational paralinguistics challenge: COVID-19 cough, COVID-19 speech, escalation & primates. In: Proc. of the 22nd annual conference of the international speech communication association (INTERSPEECH). 2021, p. 431–5.

[16] Imran A, Posokhova I, Qureshi HN, Masood U, Riaz MS, Ali K, John CN, Hussain I, Nabeel M. AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. Inform Med Unlocked 2020;20:1–14.

[17] Hassan A, Shahin I, Alsabek MB. COVID-19 detection system using recurrent neural networks. In: Proc. of the 2020 international conference on communications, computing, cybersecurity, and informatics. CCCI, 2020, p. 1–5.

[18] Laguarta J, Hueto F, Subirana B. COVID-19 artificial intelligence diagnosis using only cough recordings. IEEE Open J Eng Med Biol 2020;1:275–81.

[19] Bansal V, Pahwa G, Kannan N. Cough classification for COVID-19 based on audio MFCC features using convolutional neural networks. In: Proc. of the 2020 IEEE international conference on computing, power and communication technologies. GUCON, 2020, p. 604–8.

[20] Melek M. Diagnosis of COVID-19 and non-COVID-19 patients by classifying only a single cough sound. Neural Comput Appl 2021;33(24):17621–32.

[21] Xia T, Han J, Qendro L, Dang T, Mascolo C. Uncertainty-aware COVID-19 detection from imbalanced sound data. In: Proc. of the 22nd annual conference of the international speech communication association (INTERSPEECH). 2021, p. 2951–5.

[22] Pahar M, Klopper M, Warren R, Niesler T. COVID-19 cough classification using machine learning and global smartphone recordings. Comput Biol Med 2021;135:104572.

[23] Despotovic V, Ismael M, Cornil M, Call RM, Fagherazzi G. Detection of COVID-19 from voice, cough and breathing patterns: Dataset and preliminary results. Comput Biol Med 2021;(138):104944.

[24] Dash TK, Mishra S, Panda G, Satapathy SC. Detection of COVID-19 from speech signal using bio-inspired based cepstral features. Pattern Recognit 2021;117:107999.

[25] Stasak B, Huang Z, Razavi S, Joachim D, Epps J. Automatic detection of COVID-19 based on short-duration acoustic smartphone speech analysis. J Healthc Inform Res 2021;5(2):201–17.

[26] Han J, Brown C, Chauhan J, Grammenos A, Hasthanasombat A, Spathis D, Xia T, Cicuta P, Mascolo C. Exploring automatic COVID-19 diagnosis via voice and symptoms from crowdsourced data. In: Proc of. the IEEE international conference on acoustics, speech and signal processing. ICASSP, 2021, p. 8328–32.

[27] Coppock H, Gaskell A, Tzirakis P, Baird A, Jones L, Schuller B. End-to-end convolutional neural network enables COVID-19 detection from breath and cough audio: a pilot study. BMJ Innov 2021;7(2).

[28] Fakhry A, Jiang X, Xiao J, Chaudhari G, Han A. A multi-branch deep learning network for automated detection of COVID-19. In: Proc. of the 22nd annual conference of the international speech communication association (INTERSPEECH). 2021, p. 4139–43.

[29] Das RK, Madhavi MC, Li H. Diagnosis of COVID-19 using auditory acoustic cues. In: Proc. of the 22nd annual conference of the international speech communication association (INTERSPEECH). 2021, p. 921–5.

[30] Casanova E, Candido Jr A, Corso Fernandes Junior R, Finger M, Stefanel Gris LR, Antonelli Ponti M, Peixoto Pinto da Silva D. Transfer learning and data augmentation techniques to the COVID-19 identification tasks in ComParE 2021. In: Proc. of the 22nd annual conference of the international speech communication association (INTERSPEECH). 2021, p. 446–50.

[31] Deshpande G, Schuller BW. The DiCOVA 2021 challenge – an encoder-decoder approach for COVID-19 recognition from coughing audio. In: Proc. of the 22nd annual conference of the international speech communication association (INTERSPEECH). 2021, p. 931–5.

[32] Alkhodari MA, Khandoker AH. Detection of COVID-19 in smartphone-based breathing recordings: a pre-screening deep learning tool. PLOS ONE 2022;17(1):1–25.

[33] Dentamaro V, Giglio P, Impedovo D, Moretti L, Pirlo G. AUCO ResNet: an end-to-end network for COVID-19 pre-screening from cough and breath. Pattern Recognit 2022;127:108656.

[34] Tena A, Clarià F, Solsona F. Automated detection of COVID-19 cough. Biomed Signal Process Control 2022;71(Part):103175.

[35] Chang Y, Jing X, Ren Z, Schuller BW. CovNet: A transfer learning framework for automatic COVID-19 detection from crowd-sourced cough sounds. Front Digit Health 2021;3:799067.

[36] Nassif AB, Shahin I, Bader M, Hassan A, Werghi N. COVID-19 detection systems using deep-learning algorithms based on speech and image data. Mathematics 2022;10(4):564.

[37] Aly M, Rahouma KH, Ramzy SM. Pay attention to the speech: COVID-19 diagnosis using machine learning and crowdsourced respiratory and speech recordings. Alex Eng J 2022;61(5):3487–500.

[38] Han J, Xia T, Spathis D, Bondareva E, Brown C, Chauhan J, Dang T, Grammenos A, Hasthanasombat A, Floto A, et al. Sounds of COVID-19: exploring realistic performance of audio-based digital testing. NPJ Digit Med 2022;5(1):1–9.

[39] Sills J, Barton CM, Alberti M, Ames D, Atkinson J-A, Bales J, Burke E, Chen M, Diallo SY, Earn DJD, Fath B, Feng Z, Gibbons C, Hammond R, Heffernan J, Houser H, Hovmand PS, Kopainsky B, Mabry PL, Mair C, Meier P, Niles R, Nosek B, Osgood N, Pierce S, Polhill JG, Prosser L, Robinson E, Rosenzweig C, Sankaran S, Stange K, Tucker G. Call for transparency of COVID-19 models. Science 2020;368(6490):482–3.

[40] Bagnall AJ, Lines J, Hills J, Bostrom A. Time-series classification with COTE: The Collective of Transformation-based Ensembles. In: Proc. of the 32nd IEEE international conference on data engineering (ICDE). 2016, p. 1548–9.

[41] Pasos Ruiz A, Flynn M, Large J, Middlehurst M, Bagnall AJ. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min Knowl Discov 2021;35(2):401–49.

[42] Kakizawa Y, Shumway R, Taniguchi M. Discrimination and clustering for multivariate time series. J Amer Statist Assoc 1998;93(441):328–40.

[43] Kudo M, Toyama J, Shimbo M. Multidimensional curve classification using passing-through regions. Pattern Recognit Lett 1999;20(11):1103–11.

[44] Caiado J, Crato N, Peña D. A periodogram-based metric for time series classification. Comput Statist Data Anal 2006;50(10):2668–84.

[45] Fulcher BD, Jones NS. Highly comparative feature-based time-series classification. IEEE Trans Knowl Data Eng 2014;26(12):3026–37.

[46] Moskovitch R, Shahar Y. Classification-driven temporal discretization of multivariate time series. Data Min Knowl Discov 2015;29(4):871–913.

[47] Lines J, Bagnall A. Time series classification with ensembles of elastic distance measures. Data Min Knowl Discov 2015;29(3):565–92.

[48] Tan P, Steinbach MS, Kumar V. Introduction to data mining. Addison-Wesley; 2005.

[49] Han J, Kamber M, Pei J. Data mining: concepts and techniques. 3rd ed.. Morgan Kaufmann; 2011.

[50] Malhotra P, T.V. V, Vig L, Agarwal P, Shroff GM. TimeNet: Pre-trained deep recurrent neural network for time series classification. In: Proc. of the 25th european symposium on artificial neural networks. ESANN, 2017, p. 607–12.

[51] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Advances in neural information processing systems 27: annual conference on neural information processing systems 2014, December 8-13 2014, Montreal, Quebec, Canada. 2014, p. 3104–12.

[52] Wang Z, Yan W, Oates T. Time series classification from scratch with deep neural networks: A strong baseline. In: Proc. of the 2017 international joint conference on neural networks. IJCNN, 2017, p. 1578–85.

[53] Fawaz H, Forestier G, Weber J, Idoumghar L, Muller P. Deep learning for time series classification: a review. Data Min Knowl Discov 2019;33(4):917–63.

[54] Längkvist M, Karlsson L, Loutfi A. A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognit Lett 2014;42:11–24.

[55] Diez JR, González CA, Boström H. Boosting interval based literals. Intell Data Anal 2001;5(3):245–62.

[56] Schapire RE. A brief introduction to boosting. In: Proc. of the 16th international joint conference on artificial intelligence. IJCAI, 1999, p. 1401–6.

[57] Geurts P. Pattern extraction for time series classification. In: Principles of data mining and knowledge discovery. Springer; 2001, p. 115–27.

[58] Yamada Y, Suzuki E, Yokoi H, Takabayashi K. Decision-tree induction from time-series data based on a standard-example split test. In: Proc. of the 12th international conference on machine learning. ICML, 2003, p. 840–7.

[59] Shokoohi-Yekta M, Wang J, Keogh E. 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. SDM, 2015, p. 289–97.

[60] Balakrishnan S, Madigan D. Decision Trees for Functional Variables. In: Proc. of the 6th international conference on data mining (ICDM. 2006, p. 798–802.

[61] Bartocci E, Bortolussi L, Sanguinetti G. Data-driven statistical learning of temporal logic properties. In: Proc. of the 12th international conference on formal modeling and analysis of timed systems. FORMATS, Lecture notes in computer science, vol. 8711, Springer; 2014, p. 23–37.

[62] Baydogan MG, Runger GC. Learning a symbolic representation for multivariate time series classification. Data Min Knowl Discov 2015;29(2):400–22.

[63] Bombara G, Vasile C, Penedo F, Yasuoka H, Belta C. A decision tree approach to data classification using signal temporal logic. In: Proc. of the 19th international conference on hybrid systems: computation and control. HSCC, 2016, p. 1–10.

[64] Ye L, Keogh EJ. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Min Knowl Discov 2011;22(1–2):149–82.

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

[66] Goranko V, Montanari A, Sciavicco G. A road map of interval temporal logics and duration calculi. J Appl Non-Class Logics 2004;14(1–2):9–54.

[67] Halpern J, Shoham Y. A propositional modal logic of time intervals. J ACM 1991;38(4):935–62.

[68] Allen JF. Maintaining knowledge about temporal intervals. Commun ACM 1983;26(11):832–43.

[69] Bozzelli L, Molinari A, Montanari A, Peron A, Sala P. Model checking for fragments of the interval temporal logic HS at the low levels of the polynomial time hierarchy. Inform Comput 2018;262(Part):241–64. http://dx.doi.org/10.1016/j.ic.2018.09.006.

[70] Bozzelli L, Molinari A, Montanari A, Peron A. Model checking interval temporal logics with regular expressions. Inform Comput 2020;272:104498. http://dx.doi.org/10.1016/j.ic.2019.104498.

[71] Lubba C, Sethi S, Knaute P, Schultz S, Fulcher B, Jones N. Catch22: Canonical time-series characteristics - selected through highly comparative time-series analysis. Data Min Knowl Discov 2019;33(6):1821–52.

[72] Belson WA. A technique for studying the effects of television broadcast. J R Stat Soc 1956;5(3):195–202.

[73] Morgan JN, Sonquist JA. Problems in the analysis of survey data, and a proposal. J Amer Statist Assoc 1963;58(302):415–34.

[74] Messenger R, Mandell L. A modal search technique for predictive nominal scale multivariate analysis. J Amer Statist Assoc 1972;67(340):768–72.

[75] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. Wadsworth Publishing Company; 1984.

[76] Quinlan JR. Induction of decision trees. Mach Learn 1986;1:81–106.

[77] Quinlan JR. C4.5: Programs for machine learning. Morgan Kaufmann; 1993.

[78] Hyafil L, Rivest RL. Constructing optimal binary decision trees is NP-Complete. Inform Process Lett 1976;5(1):15–7.

[79] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.

[80] Witten IH, Frank E, Hall MA. Data mining: practical machine learning tools and techniques. 4th ed.. Morgan Kaufmann; 2017.

[81] Sadeghi B. DecisionTree.jl. 2013, https://github.com/JuliaAI/DecisionTree.jl.

[82] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. SIAM Rev 2017;59(1):65–98.

[83] Pagliarini G, Manzella F, Sciavicco G, Stan IE. ModalDecisionTrees.jl: Interpretable models for native time-series & image classification. 2021, http://dx.doi.org/10.5281/zenodo.7040419.

[84] Ho TK. Random decision forests. In: Proc. of the 3rd international conference on document analysis and recognition. ICDAR, 1995, p. 278–82.

[85] Liaw A, Wiener M. Classification and regression by RandomForest. R News 2002;2(3):18–22.

[86] Tüysüzoğlu G, Birant D, Kıranoğlu V. Temporal bagging: a new method for time-based ensemble learning. Turk J Electr Eng Comput Sci 2022;30:279–94.

[87] Pagliarini G, Sciavicco G, Stan IE. Multi-frame modal symbolic learning. In: Proc. of the 3rd workshop on artificial intelligence and formal verification, logic, automata, and synthesis (OVERLAY). CEUR workshop proceedings, vol. 2987, CEUR-WS.org; 2021, p. 37–41.

[88] Davis SB, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Process 1980;28(4):357–66.

[89] Korpáš J, Sadloňová J, Vrabec M. Analysis of the cough sound: an overview. Pulmon Pharmacol 1996;9(5–6):261–8.

[90] Singh VP, Rohith J, Mittal VK. Preliminary analysis of cough sounds. In: Proc. of the annual IEEE india conference. INDICON, 2015, p. 1–6.

[91] Friedman M. A comparison of alternative tests of significance for the problem of m rankings. Ann Math Stat 1940;11(1):86–92.

[92] Wilcoxon F. Individual comparisons by ranking methods. In: Breakthroughs in statistics. Springer; 1992, p. 196–202.

[93] Holm S. A simple sequentially rejective multiple test procedure. Scand J Stat 1979;65–70.

[94] Demšar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 2006;7(1):1–30.

[95] Benavoli A, Corani G, Mangili F. Should we really use post-hoc tests based on mean-ranks? J Mach Learn Res 2016;17(1):152–61.