Neural-Symbolic Temporal Decision Trees for Multivariate Time Series Classification

Giovanni Pagliarini[1,2] giovanni.pagliarini@unife.it
Simone Scaboro[3] scaboro.simone@spes.uniud.it
Giuseppe Serra[3] giuseppe.serra@uniud.it
Guido Sciavicco[1] guido.sciavicco@unife.it
Ionel Eduard Stan[1,2] ioneleduard.stan@unife.it

[1] Applied Computational Logic and Artificial Intelligence Laboratory (ACLAI-Lab),
Department of Mathematics and Computer Science,
University of Ferrara, Italy

[2] Department of Mathematical, Physical and Computer Sciences,
University of Parma, Italy,

[3] Department of Mathematical, Physical and Computer Sciences,
University of Udine, Italy

Artificial Intelligence

Machine Learning

Knowledge extraction from data

Functional Learning

Models ≡ mathematical functions

- Linear Regression
- k-Nearest Neighbors
- **Neural Networks**

\+ good with complex data

\+ generalization power

Symbolic Learning

Models ≡ formulas of logical languages

- Inductive Logic Programming
- Rule-Based Modeling
- **Decision Trees**

\+ good with small datasets

\+ transparency

Introduction

Artificial Intelligence

Machine Learning

KNOWLEDGE EXTRACTION FROM DATA

Functional Learning

MODELS ≡ MATHEMATICAL FUNCTIONS

- Linear Regression
- k-Nearest Neighbors
- **Neural Networks**

+ good with complex data
+ generalization power

Symbolic Learning

MODELS ≡ FORMULAS OF LOGICAL LANGUAGES

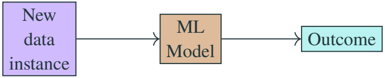
- Inductive Logic Programming
- Rule-Based Modeling
- **Decision Trees**

+ good with small datasets
+ transparency

?

Machine Learning: An Example

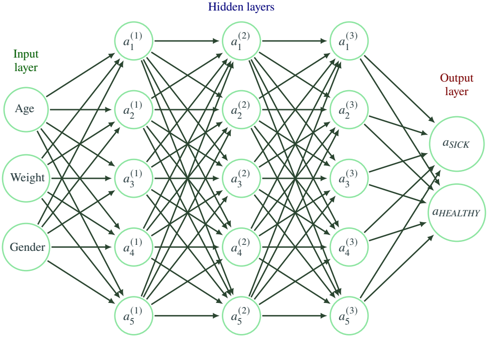
By means of a machine learning algorithm, a model can be *trained* from a dataset. The model can then, be used for prediction on new data:

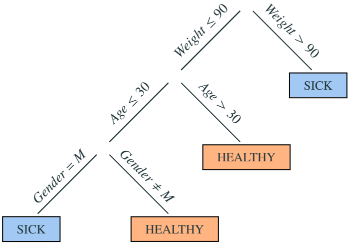


Hospitalization context where patients are *instances* described by their *age*, *weight* and *gender*. Data are normally collected in tabular form:

# Instance	Age	Weight	Gender	Class label
1	37	70	M	SICK
2	49	81	M	HEALTHY
3	20	55	F	SICK
...	...	...	...	...

Neural Networks vs. *Decision Trees* for classification





Features:

- *fluid* information flow
- *black-box* behaviour

Training algorithm:

- *fix* structure
- *initialize* parameters randomly
- iteratively *optimize* until certain conditions are met

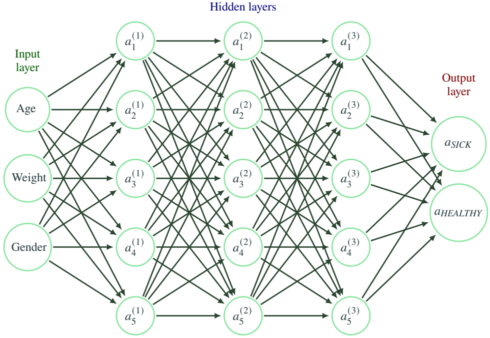
Features:

- *compact* information flow
- transparent logical reasoning

Training algorithm:

- *initialize root node* as leaf
- *find best* splitting condition
- split dataset and *recurse on children* until certain conditions are met

Neural Networks vs. *Decision Trees* for classification

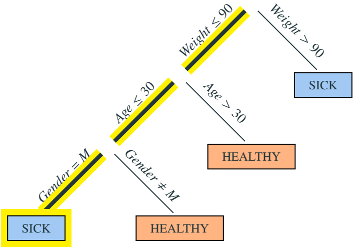


FEATURES:

- *fluid* information flow
- *black-box* behaviour

TRAINING ALGORITHM:

- *fix* structure
- *initialize* parameters randomly
- iteratively *optimize* until certain conditions are met



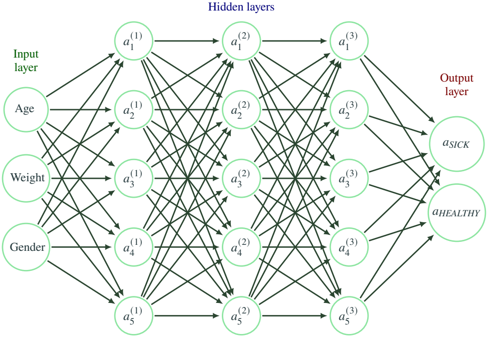
FEATURES:

- *compact* information flow
- transparent logical reasoning

TRAINING ALGORITHM:

- *initialize root node* as leaf
- *find best* splitting condition
- split dataset and *recurse on children* until certain conditions are met

Neural Networks vs. *Decision Trees* for classification

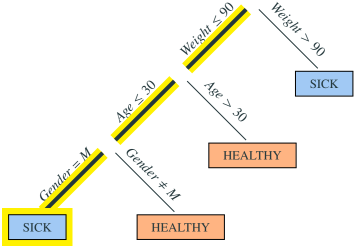


FEATURES:

- *fluid* information flow
- *black-box* behaviour

TRAINING ALGORITHM:

- *fix* structure
- *initialize* parameters randomly
- iteratively *optimize* until certain conditions are met



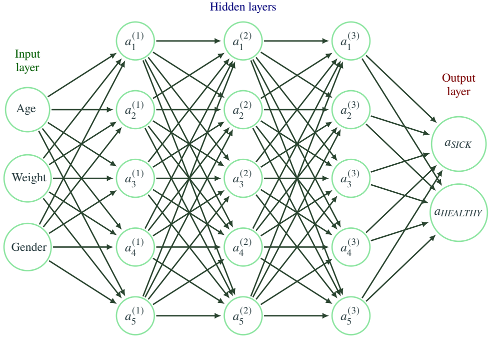
FEATURES:

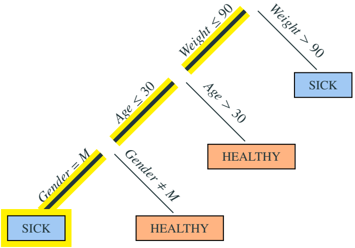
- *compact* information flow
- transparent logical reasoning

TRAINING ALGORITHM:

- *initialize root node* as leaf
- *find best* splitting condition
- split dataset and *recurse on children* until certain conditions are met

Neural Networks vs. *Decision Trees* for classification





Features:

- *fluid* information flow
- *black-box* behaviour

Training algorithm:

- *fix* structure
- *initialize* parameters randomly
- iteratively *optimize* until certain conditions are met

Features:

- *compact* information flow
- transparent logical reasoning

Training algorithm:

- *initialize root node* as leaf
- *find best* splitting condition
- split dataset and *recurse on children* until certain conditions are met

Neuro-Symbolic strategies

- DT → NN [Sethi 1990, Brent 1991, Ivanova & Kubat 1995, Setiono & Leow 2000]
  - Train a DT
  - Map it to a NN
  - Optimize the NN

+ performance

- NN → DT [Towell & Shavlik 1993, Craven & Shavlik 1995, Krishnan et al. 1999, Zhou and Jiang 2004]
  - Train a NN
  - Map it to a DT
  - Prune the DT

+ transparency

- Define a **hybrid NN/DT model** and a **learning algorithm** for it

Neuro-Symbolic strategies

- DT $\rightarrow$ NN [Sethi 1990, Brent 1991, Ivanova & Kubat 1995, Setiono & Leow 2000]
  - Train a DT
  - Map it to a NN
  - Optimize the NN

  \+ performance
- NN $\rightarrow$ DT [Towell & Shavlik 1993, Craven & Shavlik 1995, Krishnan et al. 1999, Zhou and Jiang 2004]
  - Train a NN
  - Map it to a DT
  - Prune the DT

  \+ transparency
- Define a **hybrid NN/DT model** and a **learning algorithm** for it

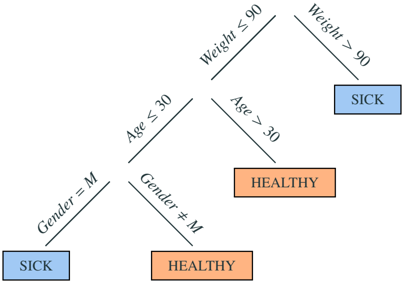
Neuro-Symbolic strategies

- DT $\longrightarrow$ NN [Sethi 1990, Brent 1991, Ivanova & Kubat 1995, Setiono & Leow 2000]
  - Train a DT
  - Map it to a NN
  - Optimize the NN

  \+ performance
- NN $\longrightarrow$ DT [Towell & Shavlik 1993, Craven & Shavlik 1995, Krishnan et al. 1999, Zhou and Jiang 2004]
  - Train a NN
  - Map it to a DT
  - Prune the DT

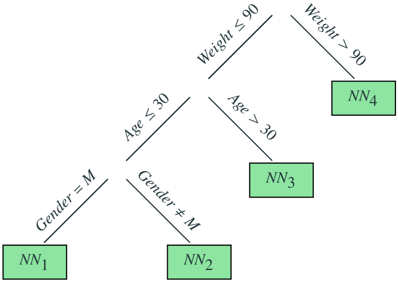
  \+ transparency
- Define a **hybrid NN/DT model** and a **learning algorithm** for it

Neuro-Symbolic strategies: NN/DT Hybrids



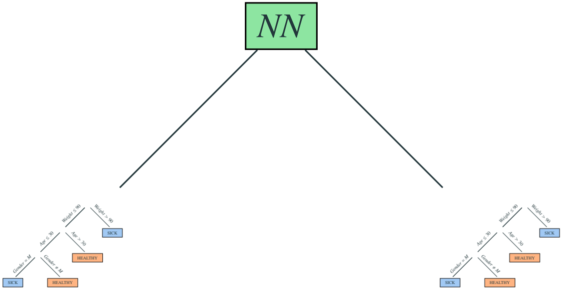
- *leaf-level* hybrid: logical reasoning + more specific neural reasoning
- *root-level* hybrid: neural reasoning + more specific logical reasoning
- *split-level* hybrid: mixed reasoning (and it still provides some form of explanation).

Neuro-Symbolic strategies: NN/DT Hybrids



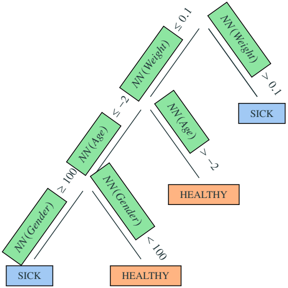
- *leaf-level* hybrid: logical reasoning + more specific neural reasoning
- *root-level* hybrid: neural reasoning + more specific logical reasoning
- *split-level* hybrid: mixed reasoning (and it still provides some form of explanation).

Neuro-Symbolic strategies: NN/DT Hybrids



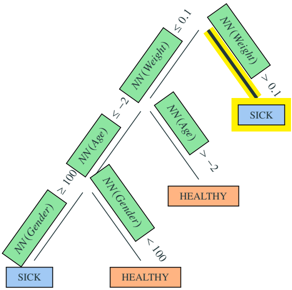
- *leaf-level* hybrid: logical reasoning + more specific neural reasoning
- *root-level* hybrid: neural reasoning + more specific logical reasoning
- *split-level* hybrid: mixed reasoning (and it still provides some form of explanation).

Neuro-Symbolic strategies: NN/DT Hybrids

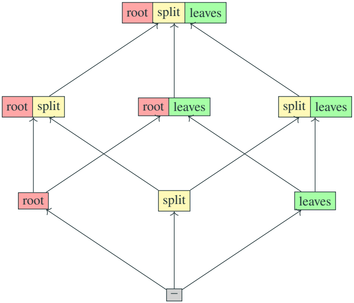


- *leaf-level* hybrid: logical reasoning + more specific neural reasoning
- *root-level* hybrid: neural reasoning + more specific logical reasoning
- *split-level* hybrid: mixed reasoning (and it still provides some form of explanation).

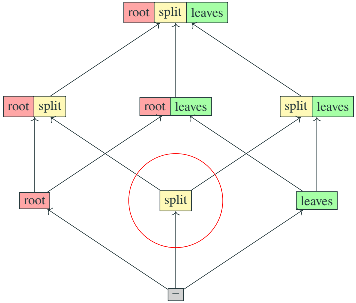
Neuro-Symbolic strategies: NN/DT Hybrids



- *leaf-level* hybrid: logical reasoning + more specific neural reasoning
- *root-level* hybrid: neural reasoning + more specific logical reasoning
- *split-level* hybrid: mixed reasoning (and it still provides some form of explanation).



- *leaf-level* hybrid: logical reasoning + more specific neural reasoning
- *root-level* hybrid: neural reasoning + more specific logical reasoning
- *split-level* hybrid: mixed reasoning (and it still provides some form of explanation).



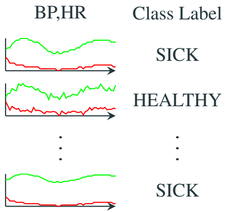
- *leaf-level* hybrid: logical reasoning + more specific neural reasoning
- *root-level* hybrid: neural reasoning + more specific logical reasoning
- ***split-level*** **hybrid**: mixed reasoning (and it still provides some form of explanation).

Multivariate Time Series Classification (MTSC)

Static Dataset

Age	Weight	Gender	Class label
37	70	M	SICK
49	81	M	HEALTHY
20	55	F	SICK
...	...	...	...

Temporal Dataset



- A multivariate time series is a set of variables that evolve through time;
- Multivariate Time Series Classification (MTSC) is an important ML task;
- Traditional, static decision trees can solve MTSC tasks in a limited way;
- Interval temporal decision trees can solve MTSC tasks with temporal reasoning.

A temporal dataset $\mathcal{T} = \{T_1, \ldots, T_m\}$ is a finite collection of *temporal instances*, each consisting of a N-point time series of n temporal variables $\mathcal{V} = \{V_1, \ldots, V_n\}$, and associated with a *class label* from a set of *classes* $\mathcal{C} = \{C_1, \ldots, C_k\}$.

Static decision trees encompass a set of split decisions, which is equal to the alphabet $\mathcal{P}$:

$$\mathcal{S} = \mathcal{P} = \{f(V) \bowtie v \mid V \in \mathcal{V}, v \in dom(f)\},$$

where f is a *scalar feature function*, and $\bowtie \in \{\leq, =, \neq, >\}$.

Static decision trees are formulas of the following grammar, where $S \in \mathcal{S}$ and $C \in \mathcal{C}$:

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

Temporal Decision Trees for MTSC

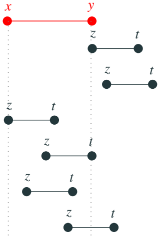
The key idea behind interval temporal decision trees is that the temporal dimension can be handled by using a temporal modal logic based on intervals ($\mathcal{HS}$, originally presented by J. Halpern and Y. Shoham). $\mathcal{HS}$ formulas are defined by the following grammar:

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

where $p \in \mathcal{P}$ is an atomic proposition, and $X \in \{A, L, B, E, D, O, \overline{A}, \overline{L}, \overline{B}, \overline{E}, \overline{D}, \overline{O}\}$ is one of the 12 binary interval relations (J.F. Allen, 1983).

Table 1: Six Allen's interval relations. Other six relations can be defined as their inverses.

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



Unlike the static case, propositions are relativized to intervals of the series, and their decisions may ask whether there exists an interval, with respect to the current one, with the given propositional property.

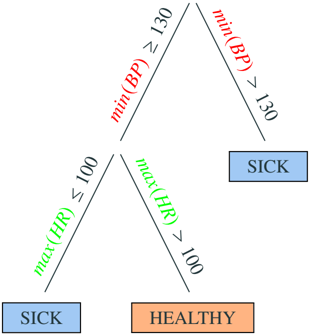
Thus, the language of temporal decision trees encompasses a set of temporal split ecisions:

$$\begin{aligned}\mathcal{S} \quad &= \quad \{f(V) \bowtie v \mid V \in \mathcal{V}, v \in dom(f)\} \cup \\ & \quad\quad \{\langle X \rangle (f(V) \bowtie v) \mid X \in \mathcal{X}, V \in \mathcal{V}, v \in dom(f)\},\end{aligned}$$

where $\mathcal{X} = \{A, L, B, E, D, O, \overline{A}, \overline{L}, \overline{B}, \overline{E}, \overline{D}, \overline{O}\}$ are interval operators of $\mathcal{HS}$.

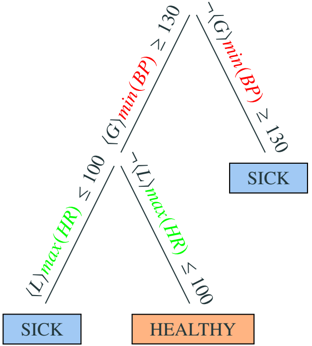
Static vs. Temporal Decision Trees

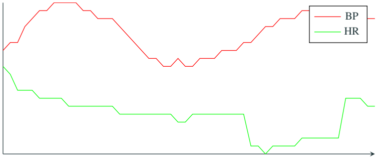
Static Decision Tree



$min(BP)$	$max(BP)$	$min(HR)$	$max(HR)$
140	160	45	90

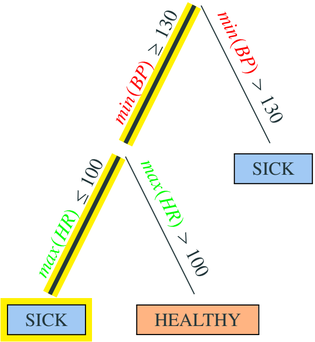
Temporal Decision Tree



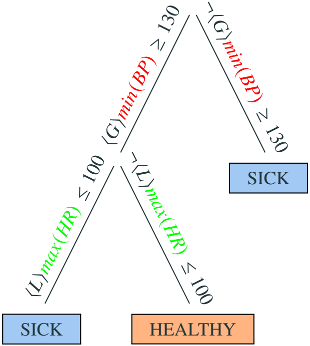


Static vs. Temporal Decision Trees

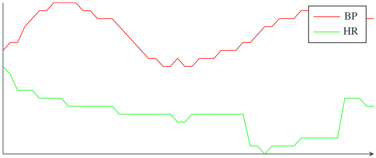
Static Decision Tree



Temporal Decision Tree

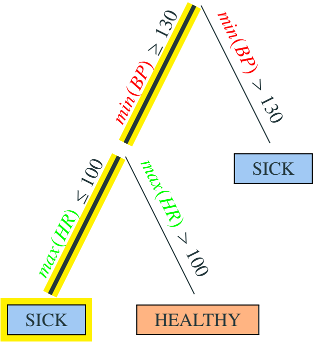


$min(BP)$	$max(BP)$	$min(HR)$	$max(HR)$
140	160	45	**90**

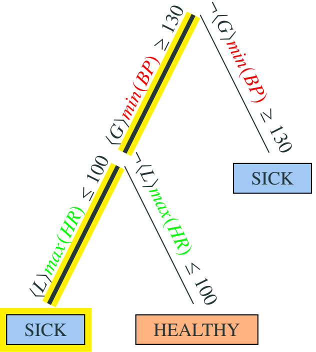


Static vs. Temporal Decision Trees

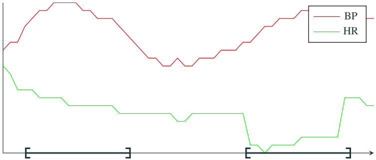
Static Decision Tree



Temporal Decision Tree



$min(BP)$	$max(BP)$	$min(HR)$	$max(HR)$
140	160	45	**90**



Autoencoders (as feature extractors)

$$\langle X \rangle (f(V) \bowtie v)$$

Autoencoder schema:

- Train a network to reproduce its input (i.e., learn the identity function);
- Introduce an information bottleneck;
- As a result, the prefix of the network (encoder) is forced to produce a succint representation of the input.

Autoencoders (as feature extractors)

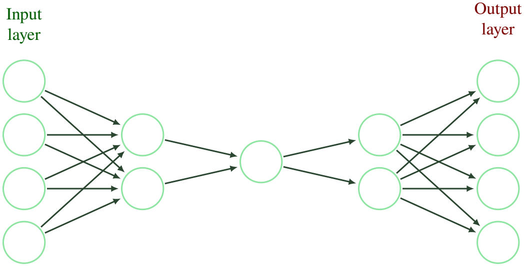
$$\langle X \rangle (\boxed{NN} (V) \bowtie v)$$

Autoencoder schema:

- Train a network to reproduce its input (i.e., learn the identity function);
- Introduce an information bottleneck;
- As a result, the prefix of the network (encoder) is forced to produce a succint representation of the input.

Autoencoders (as feature extractors)

$$\langle X \rangle (\boxed{NN} (V) \bowtie v)$$



Autoencoder schema:

- Train a network to reproduce its input (i.e., learn the identity function);
- Introduce an information bottleneck;
- As a result, the prefix of the network (encoder) is forced to produce a succint representation of the input.

With time series data, sequence-to-sequence and transformer models are commonly used. Note that they allow inputs of varying lengths.

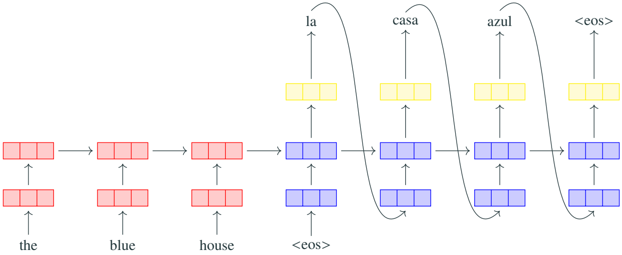


Figure 1: Example of a sequence-to-sequence model used for natural language translation.

Experiments in a cross-validation setting were done on three benchmark datasets for MTSC:

Dataset	**# train+test instances**	**# points** (N)	**# variables** (n)	**# classes** (k)
Libras	180 + 180 = 360	45	2	15
NATOPS	180 + 180 = 360	51	24	6
RacketSports	151 + 152 = 303	30	6	4

For each dataset, six approaches were compared:

- Static DT with *min* and *max*;
- Temporal DT with *min* and *max*;
- Static DT with neural features;
- Temporal DT with neural features;
- Static DT with neural features, *min* and *max*;
- Temporal DT with neural features, *min* and *max*.

Experiments in a cross-validation setting were done on three benchmark datasets for MTSC:

Dataset	**# train+test instances**	**# points** (N)	**# variables** (n)	**# classes** (k)
Libras	180 + 180 = 360	45	2	15
NATOPS	180 + 180 = 360	51	24	6
RacketSports	151 + 152 = 303	30	6	4

For each dataset, six approaches were compared:

- Static DT with *min* and *max*;
- Temporal DT with *min* and *max*;
- Static DT with neural features;
- Temporal DT with neural features;
- Static DT with neural features, *min* and *max*;
- Temporal DT with neural features, *min* and *max*.

Experiments: Statistical Results

Table 2: Average results (metrics are shown in percentage points).

			κ coeff.	accuracy	F1	time (s)
Libras	DT	min, max	35.4	39.7	39.3	0.1
		neural	19.0	24.4	23.8	0.1
		min, max, neural	40.9	44.8	44.2	0.1
	TDT	min, max	54.6	57.6	57.2	6.3
		neural	54.5	57.5	56.7	18.0
		min, max, neural	**55.2**	**58.2**	**57.6**	**30.7**
NATOPS	DT	min, max	65.1	70.9	70.8	0.7
		neural	42.8	52.3	52.1	0.6
		min, max, neural	65.7	71.5	71.4	1.0
	TDT	min, max	84.0	86.7	86.7	37.0
		neural	**87.1**	**89.2**	**89.3**	**118.3**
		min, max, neural	86.7	88.9	89.0	252.1
RacketSports	DT	min, max	55.4	66.6	67.4	0.2
		neural	44.2	58.4	59.2	0.2
		min, max, neural	**57.5**	**68.2**	**69.3**	**0.3**
	TDT	min, max	55.0	66.3	67.5	1.1
		neural	56.0	67.1	68.1	2.7
		min, max, neural	56.3	67.3	68.3	5.5

The best approach for each dataset involves neural features.

Table 3: Per-class recall in percentage points.

		Libras															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	AA
DT	min, max	29	41	26	45	53	38	34	39	**88**	28	38	38	36	16	**48**	40
	neural	17	12	13	46	38	22	34	8	30	14	28	32	19	22	30	24
	min, max, neural	35	57	25	59	53	31	42	50	87	36	41	43	44	28	42	45
TDT	min, max	48	77	**49**	70	**70**	49	**64**	55	**88**	**42**	**55**	51	41	**58**	**48**	58
	neural	**59**	76	42	68	59	**56**	62	**58**	**88**	35	52	**60**	**50**	50	47	58
	min, max, neural	56	**78**	46	**72**	69	51	62	55	86	39	54	53	53	52	47	**58**

		NATOPS						
		1	2	3	4	5	6	AA
DT	min, max	91	77	65	52	51	90	71
	neural	48	45	40	60	52	69	52
	min, max, neural	91	73	64	57	53	90	72
TDT	min, max	93	87	68	90	90	92	87
	neural	**95**	**88**	**70**	**91**	**94**	92	**89**
	min, max, neural	**95**	87	67	**91**	**94**	**94**	89

		RacketSports				
		1	2	3	4	AA
DT	min, max	50	**55**	83	84	68
	neural	54	36	85	64	60
	min, max, neural	55	52	**88**	85	**70**
TDT	min, max	51	54	80	85	68
	neural	**60**	49	77	**87**	68
	min, max, neural	54	53	82	84	69

In more than half of the classes, neural features improve the class recall.

Conclusion

- We taxonomized neural-symbolic approaches based on neural networks and decision trees;
- We extended decision trees to the use of a temporal modal logic in order to tackle MTSC tasks;
- We introduced autoencoders for achieving a feature extraction specific to each temporal variable;
- We compared the *split*-level NN/DT hybrid approach with standard approaches at decision tree modeling, that involve flattening the time axis via simple features (*minimum* and *maximum*);
- We showed that this approach improves the performance w.r.t. symbolic-only decision trees.

Future steps

- Define and experiment with the other presented neural-symbolic approaches (e.g, leaf-level);
- Investigate on the level of transparency and interpretability of these approaches.

Neural-Symbolic Temporal Decision Trees for Multivariate Time Series Classification

Giovanni Pagliarini[1,2] giovanni.pagliarini@unife.it
Simone Scaboro[3] scaboro.simone@spes.uniud.it
Giuseppe Serra[3] giuseppe.serra@uniud.it
Guido Sciavicco[1] guido.sciavicco@unife.it
Ionel Eduard Stan[1,2] ioneleduard.stan@unife.it

[1] Applied Computational Logic and Artificial Intelligence Laboratory (ACLAI-Lab), Department of Mathematics and Computer Science, University of Ferrara, Italy

[2] Department of Mathematical, Physical and Computer Sciences, University of Parma, Italy,

[3] Department of Mathematical, Physical and Computer Sciences, University of Udine, Italy