Towards hybrid fault diagnosis: machine learning enhanced with meta knowledge

Louis Goupil, **Elodie Chanthery,** Louise Travé-Massuyès,
Sébastien Delautier (ATOS)

Context

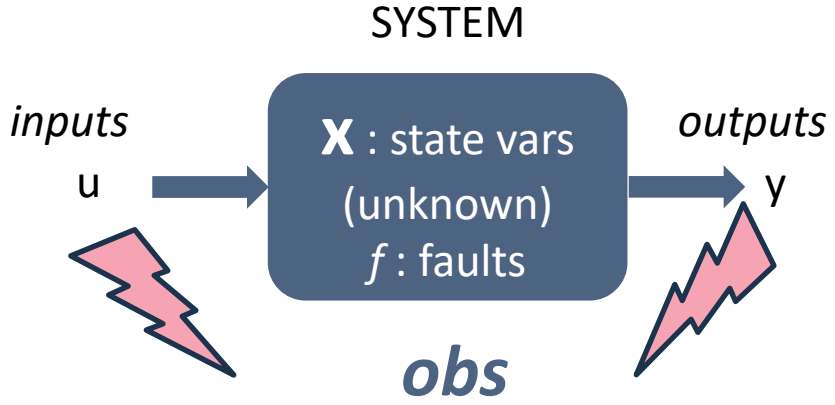
Goal: **Diagnosis** and **correction** of a system without model

Diagnosis of a system without physical model can be achieved with **Data-Based** methods

Correction of a faulty system requires **isolation/explanation** of the fault

→ **Data provides meaningful information but important knowledge can be leveraged from Model-Based Diagnosis**

Model based diagnosis knowledge



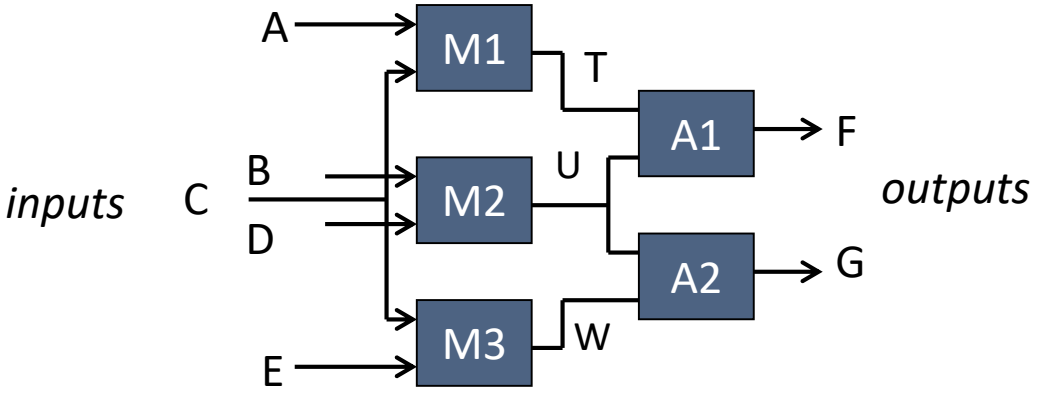
Diagnosis indicator

A relation $\mathrm{R}(\dot{x}', \ddot{x}', \ldots) = \mathrm{r(t)}$, with input x', a subvector of the observables x and output a scalar r named residual is a **diagnosis indicator** d if, for each x measured in the system under nominal conditions, the relation is fulfilled, i.e., $r = 0$ (or tends to 0 when time tends to ∞ in the case of a dynamic system)

Diagnosis indicators (Analytical Redundancy Relation):

~~**X** : state vars (unknown)~~

Model ⇨ Relations $\mathrm{R}(\boldsymbol{obs}) = \mathrm{r(t)}$ such that $\lim_{t\to\infty} r(t) = 0$ for all ***obs*** consistent with normal behavior.



Simple static polybox

~~T, U, W~~ ⇨

	ok	Faulty M1	Faulty M2	Faulty M3	Faulty A1	Faulty A2
R1 : A.C + B.D – F = 0	0	X	X	0	X	0
R2 : E.C + B.D – G = 0	0	0	X	X	0	X
R3 : A.C – E.C – F + G = 0	0	X	0	X	X	X

Problem and objective

Let $D = \{(x^i, \lambda^i)\}$ a dataset of labeled samples with

- $x^i \in \mathbb{R}^n$ a vector of observable variables
- $\lambda^i \in \Lambda = \{\lambda_k\}_{k=0\ldots n_F}$ the set of labels with λ_0 the nominal label and $\lambda_1, \ldots, \lambda_{N_F}$ the faulty labels associated to faults $F_1, \ldots F_{n_F}$

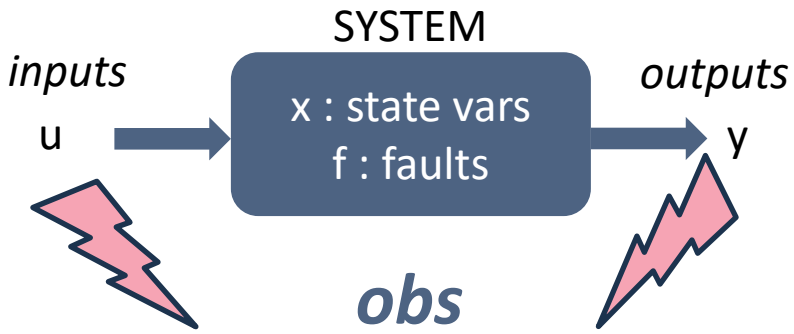
Fault Diagnosis Objective (case of maximal diagnosability*)

Based on observable variables obs, find a function that partitions D into a set of classes C so that

- Faulty samples are separated from nominal samples (detectability)
- Faulty samples labeled by λ_j are separated from faulty sampled labeled by λ_k if j $\neq k$ (isolability)

*If some faults are non detectable or non isolable → find the best solution

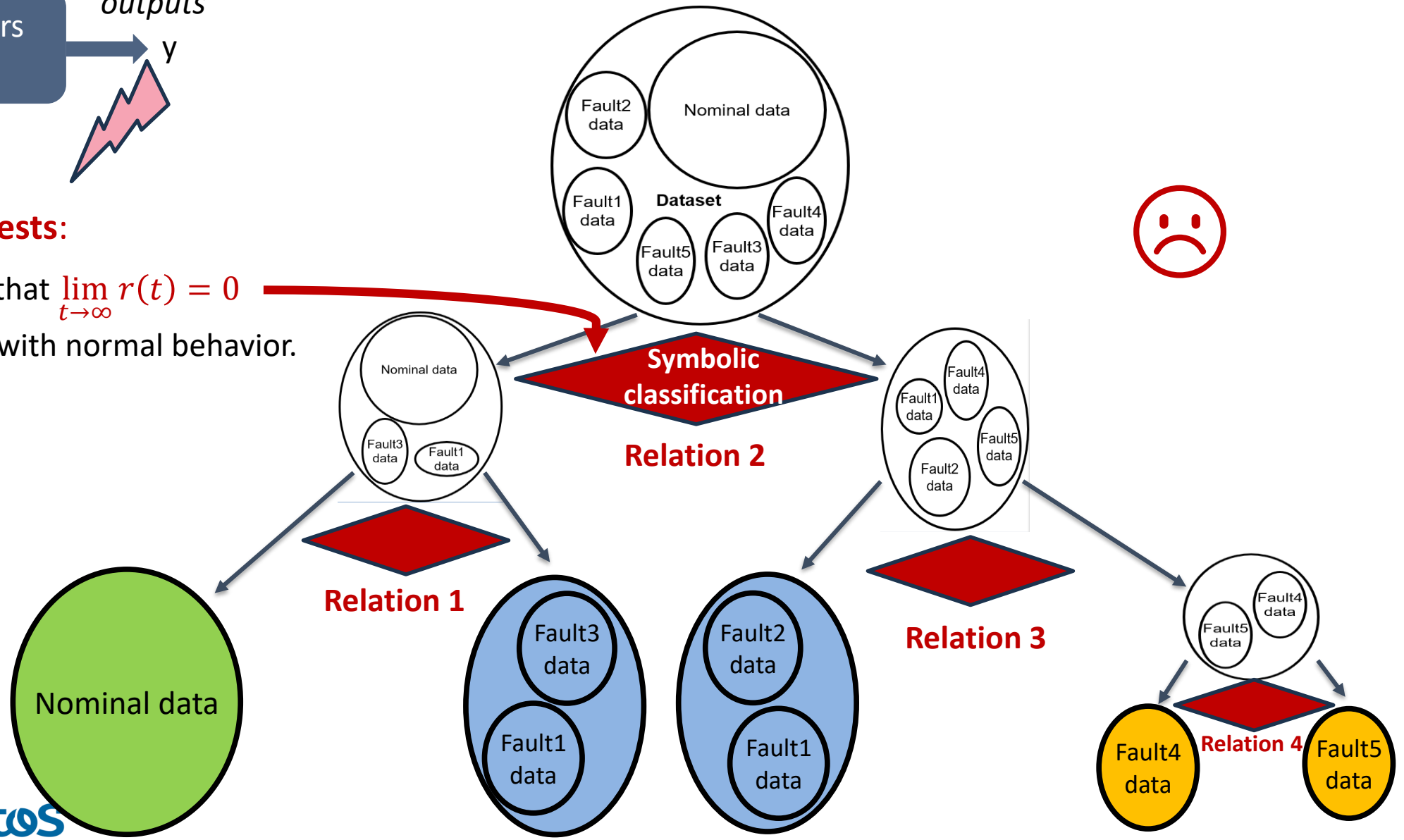
Data based diagnosis: multivariate decision trees



Diagnosis tests:

$\mathbf{R}(obs) = r(t)$ such that $\lim_{t\to\infty} r(t) = 0$

for all ***obs*** consistent with normal behavior.



DT4X – From Class Separation to Data-Based ARRs

- Other methods → seek separators between classes
- Our method → seeks relations of the form $\mathrm{R}(\boldsymbol{obs}) = 0$ that are satisfied by some classes and not others

Conceptual shift

- Inspired by ARRs for model-based diagnosis
- Nodes generate non-linear multivariate expressions representing input-output relations of the system

Interpretability & Diagnosis

- These expressions reflect system's physical reality, not just data separation
- In fault diagnosis, input-output relations are known as ARRs and used as fault indicators

→ DT4X discovers what we defined as **data-based ARRs** i.e fault indicators discovered from data

Symbolic classification

Algorithm to estimate a function $f = t \circ \hat{f}$ such as:

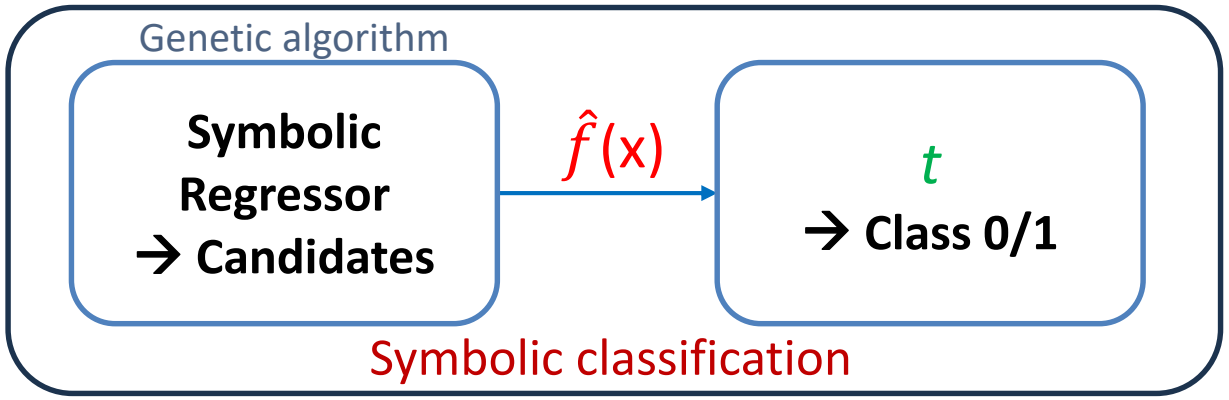
Known or chosen by the user
unknown

$$x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n \xrightarrow{\hat{f}} a \in \mathbb{R} \xrightarrow{t} label \in \{0, 1\}$$

Dataset $\mathcal{D}$ of p pairs $(x, label)$

Set of operators
*(ex: +, *, -, /, sin, arctan, ln, ²)*

Threshold ε



Evaluation of the candidate c with the log-loss fitness function

$$Fitness(\hat{f}) = -\frac{1}{p} \sum_{(x,label)\in\mathcal{D}} [label * \ln\left(t(c(x))\right) + (1 - label) * \ln\left(1 - t(c(x))\right)]$$

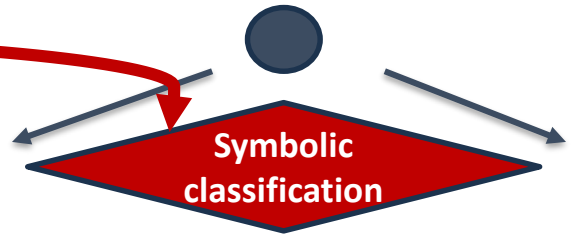
Stopping condition

IF $Fitness(\hat{f}) < \varepsilon$ **THEN** algorithm stops and f is « found » **ELSE** $Mutate(\hat{f})$ **AND** $Repeat$

DT4X: Diagnosis Tree for eXplainability

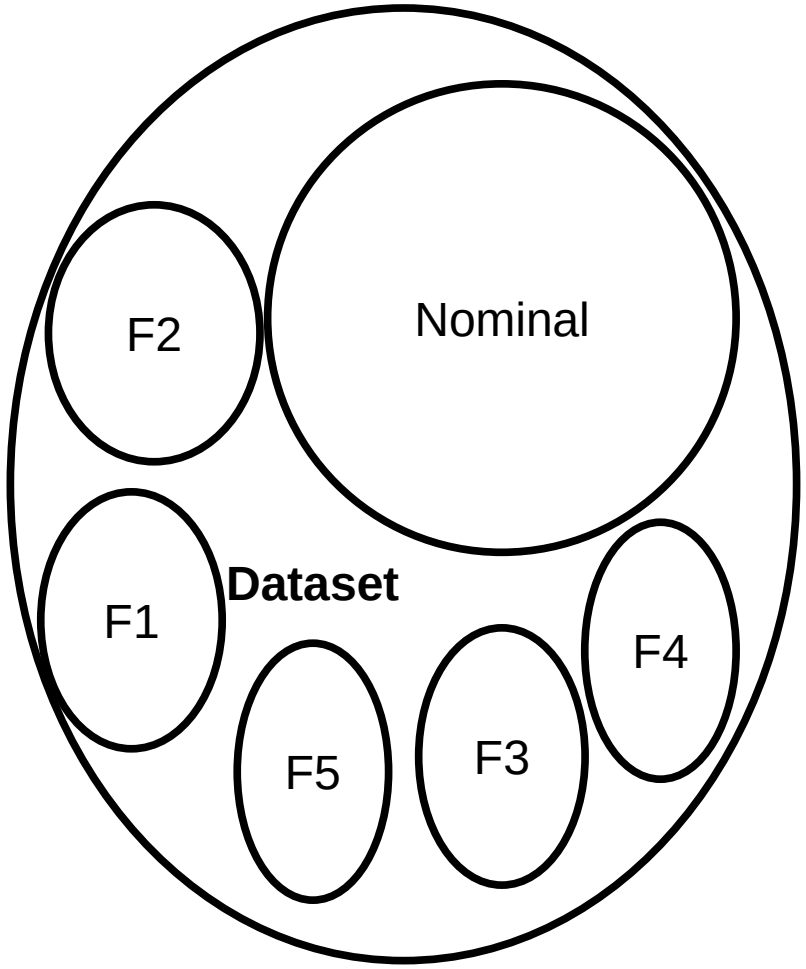
Diagnosis

Meta-knowledge



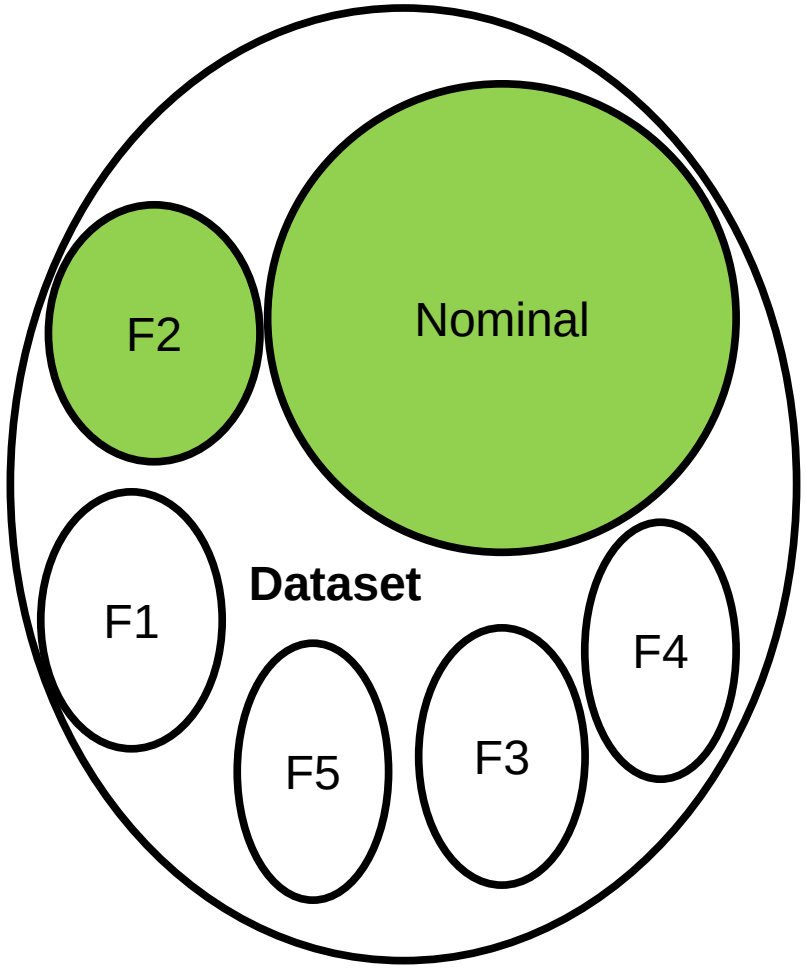
Train a symbolic classifier to find f

DT4X - Principle

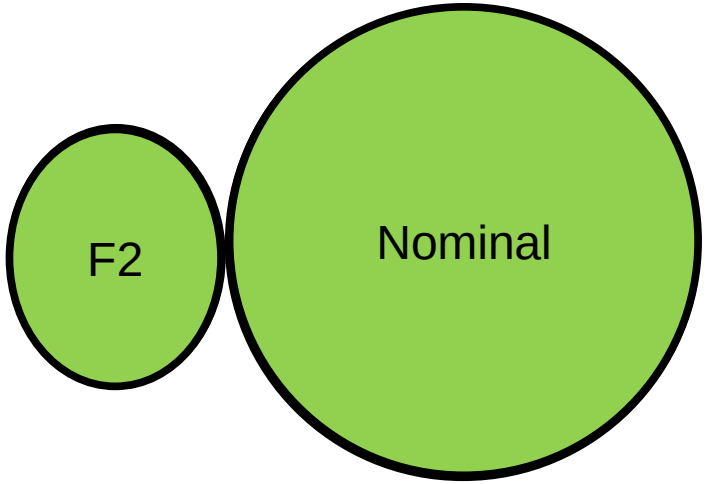


DT4X - Principle

Select 2 labels
(preferably the nominal label)



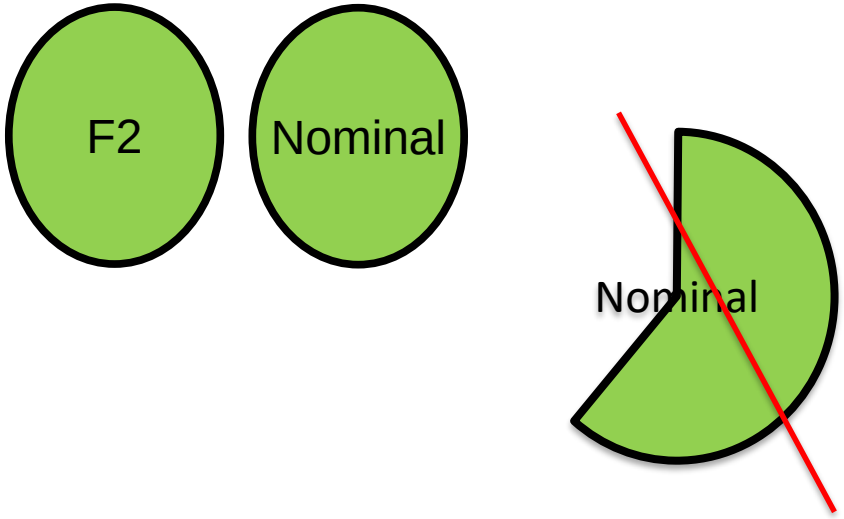
DT4X - Principle



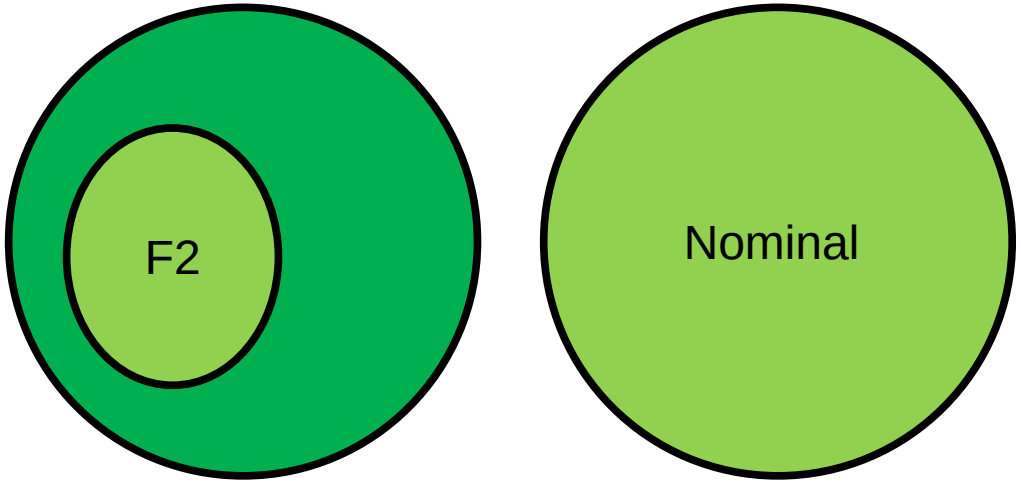
DT4X - Principle

Balancing sets

Reducing the majority label



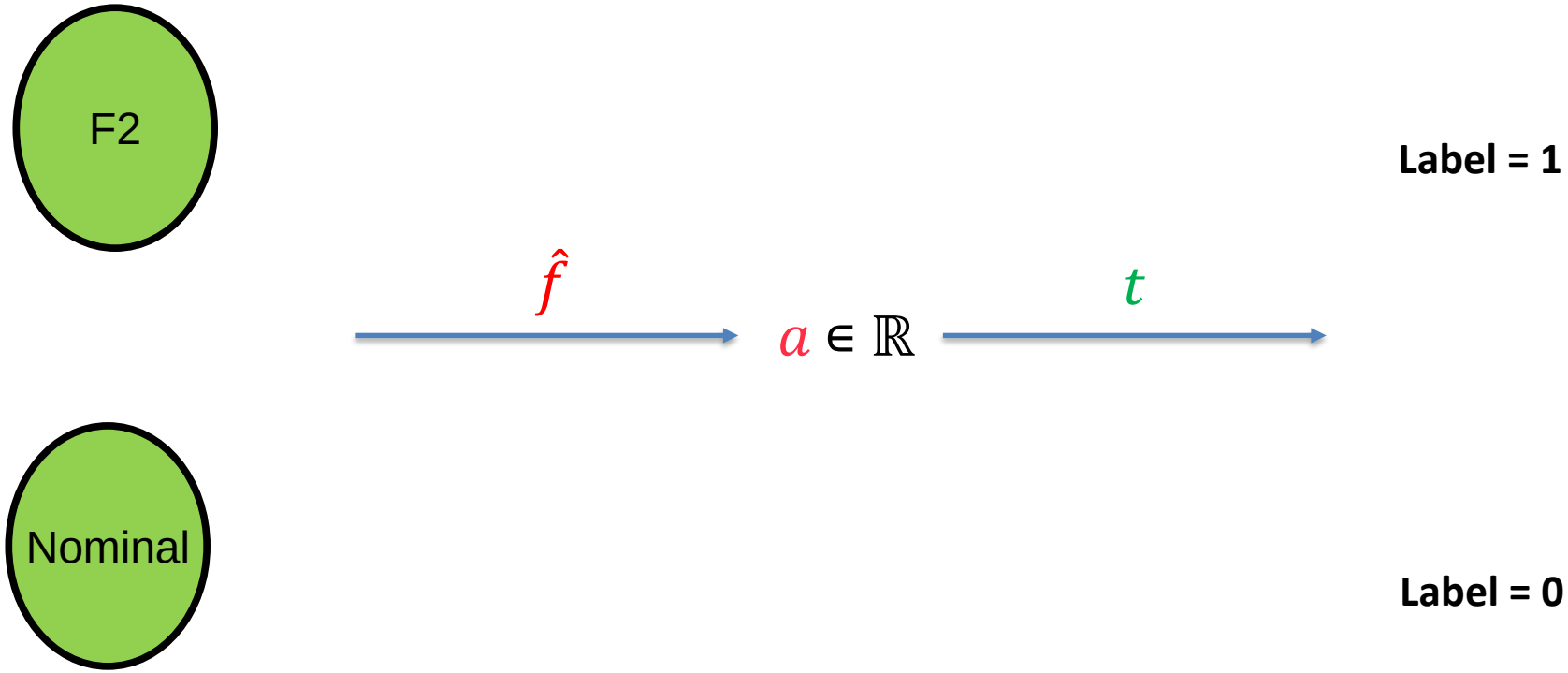
Increasing the minority label



DT4X - Principle

DT4X - Principle

Use **symbolic classification** to find $f = t \circ \hat{f}$



DT4X - Principle

F2

Nominal

$\hat{f}$ → $a \in \mathbb{R}$ → t

Label = 1

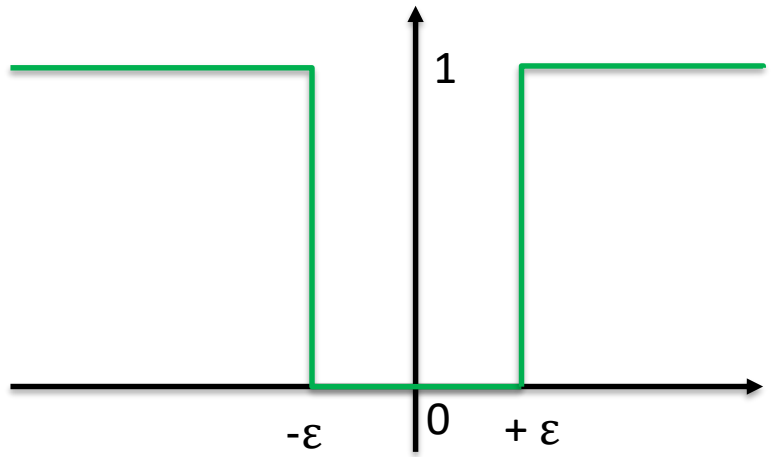
$\hat{f}(x) = a$ $\neq_\epsilon 0$

$\hat{f}(x) = a$ $=_\epsilon 0$

meta-knowledge of model-based diagnosis

Label = 0

t:



ε is a hyperparameter of DT4X

DT4X - Principle

f is found

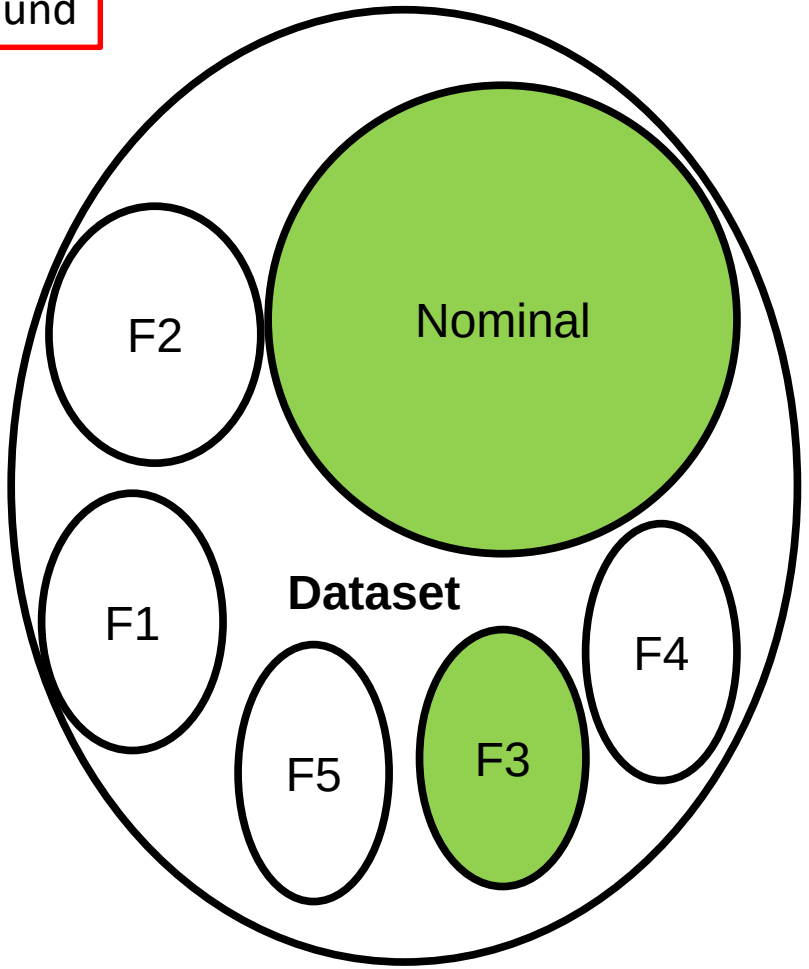
f is not found

DT4X - Principle

f is found

f is not found

Choose another pair

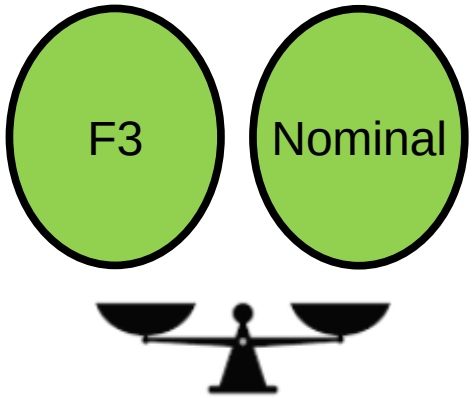


DT4X - Principle

f is found

f is not found

Repeat steps

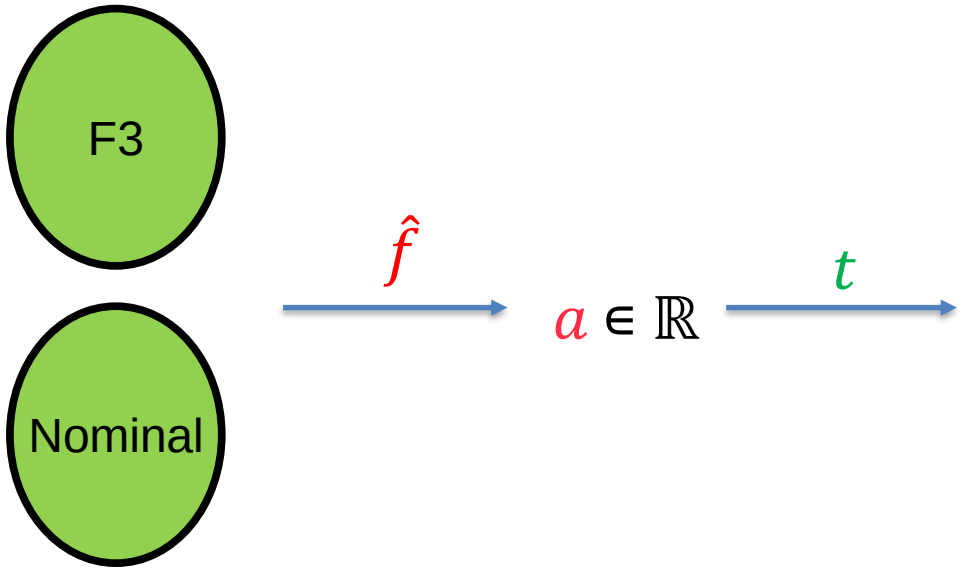


DT4X - Principle

f is found

f is not found

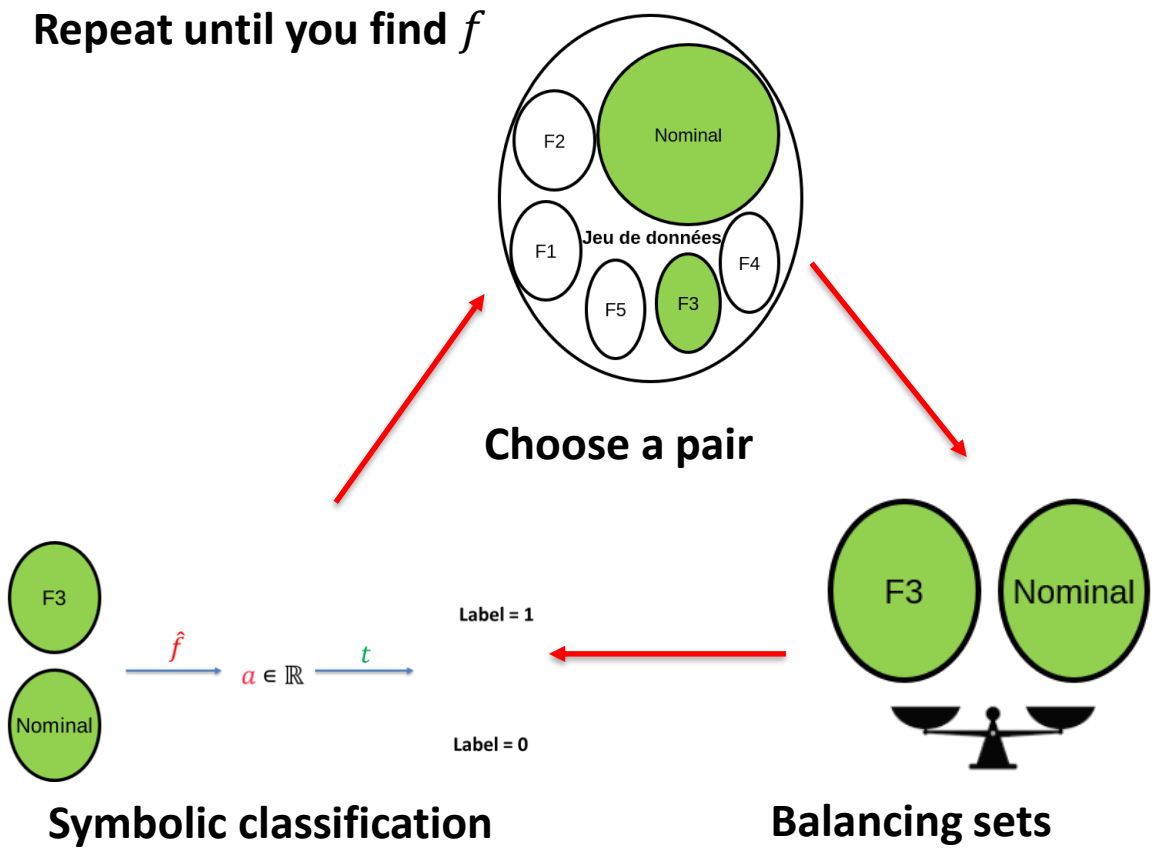
Repeat steps



DT4X - Principle

f is found

f is not found



DT4X - Principle

f is found

f is not found

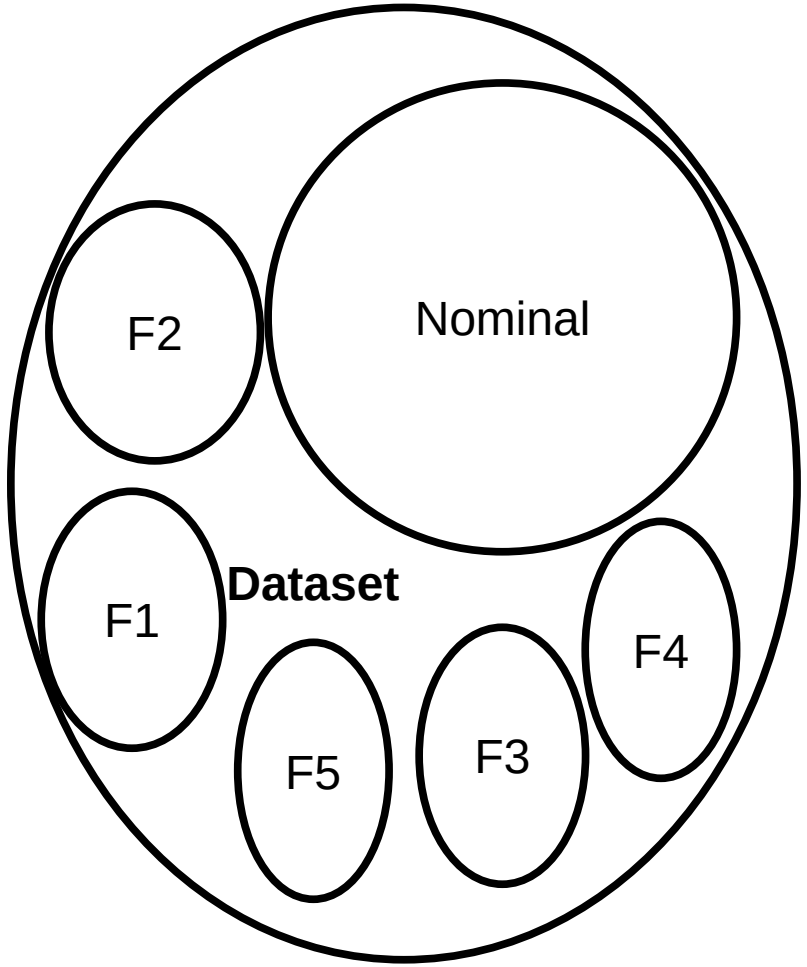
IF
all pairs have been tried on
AND
f is not found

THEN
Stop the algorithm

DT4X - Principle

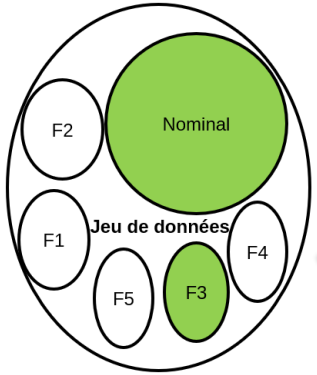
f is found

The entire dataset **is evaluated by** $t \circ \hat{f}$

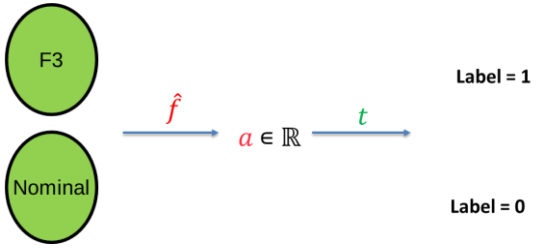


f is not found

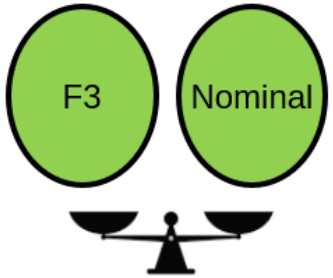
Repeat until you find f



Choose a pair



Symbolic classification

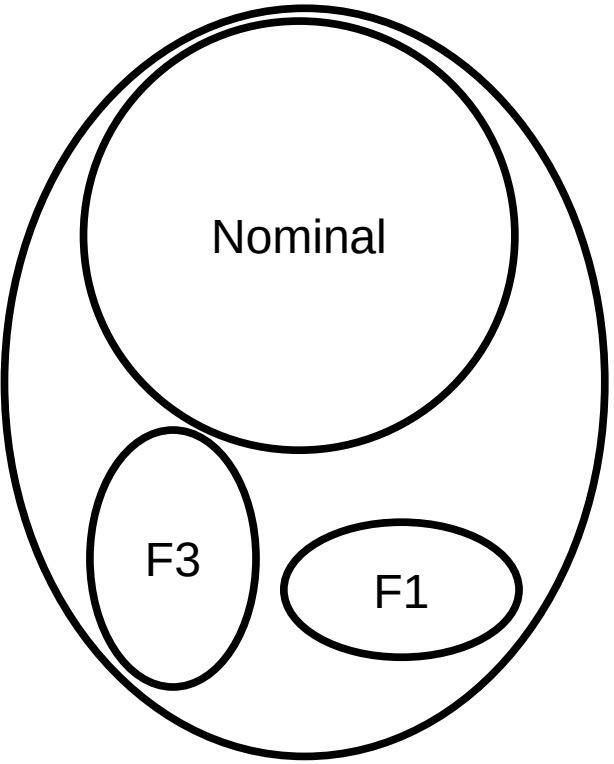


Balancing sets

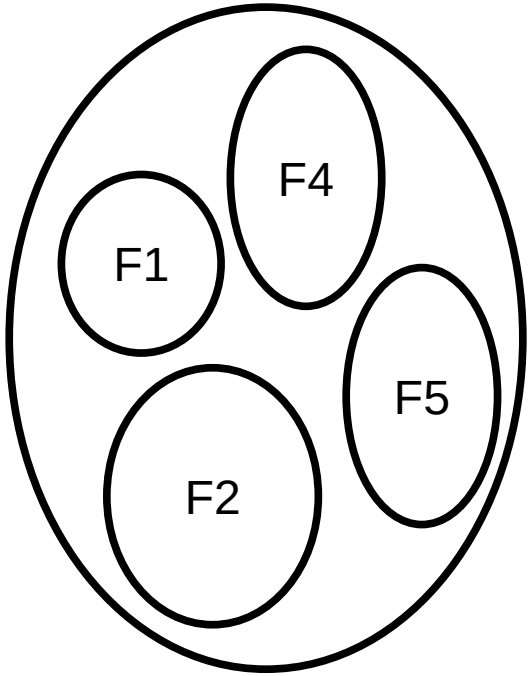
DT4X - Principle

f is found

$t \circ \hat{f} = 0$

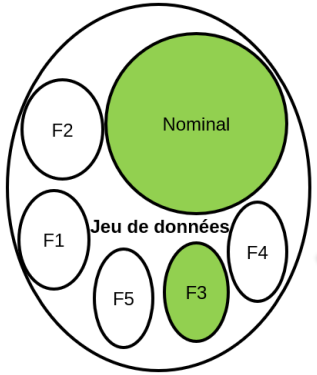


$t \circ \hat{f} = 1$



f is not found

Repeat until you find f



Choose a pair

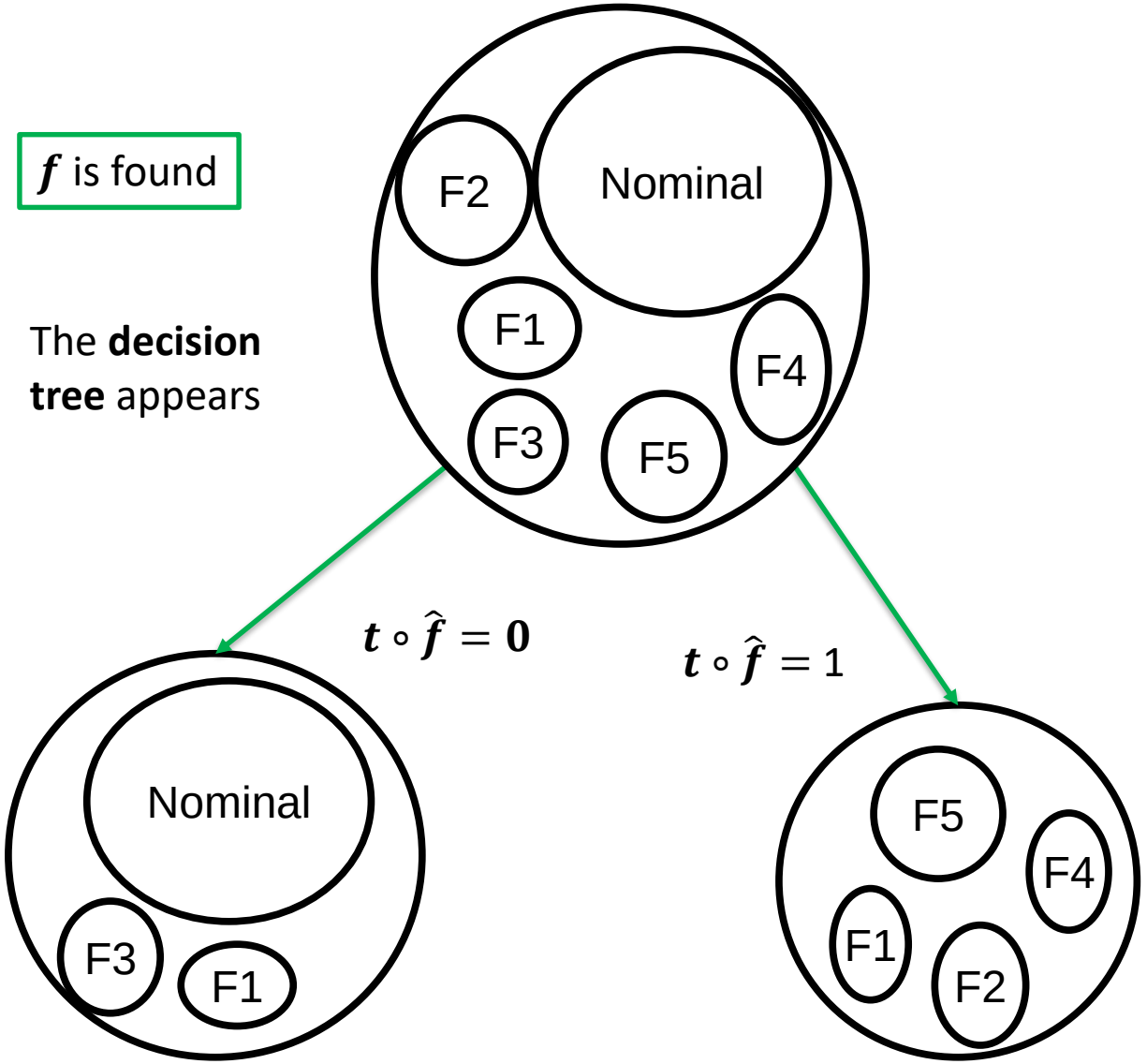
Symbolic classification

Balancing sets

DT4X - Principle

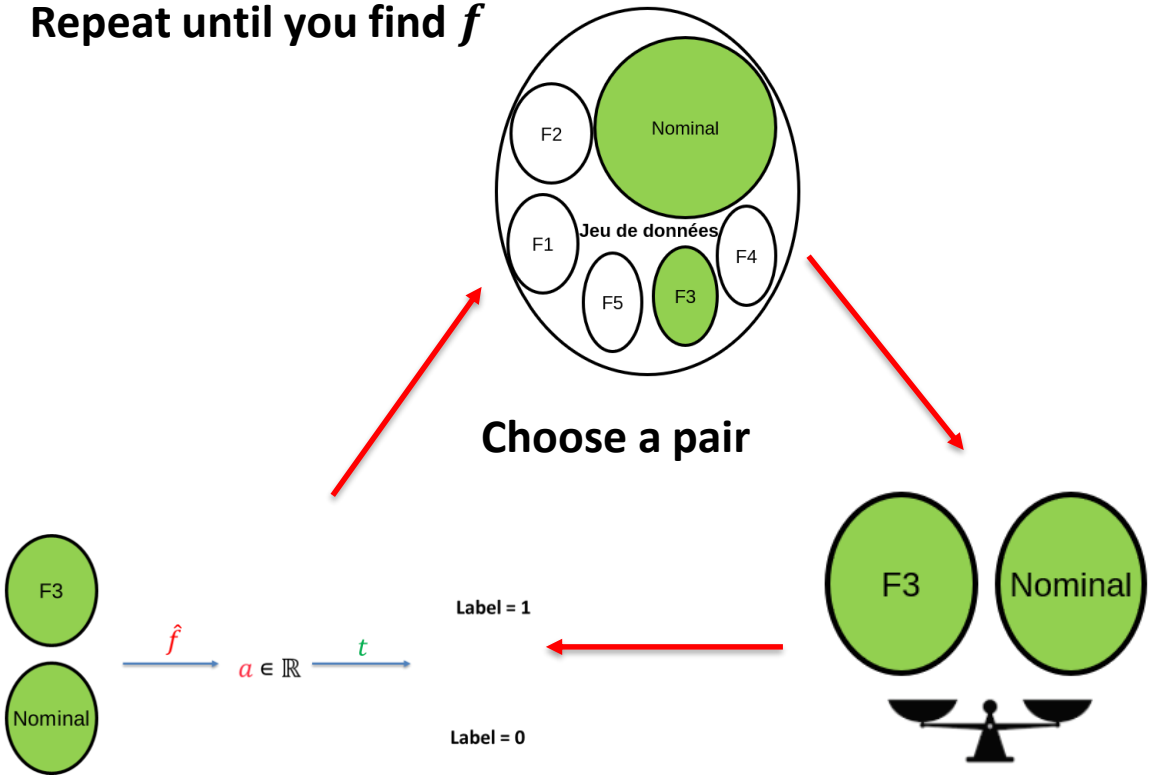
f is found

The **decision tree** appears



f is not found

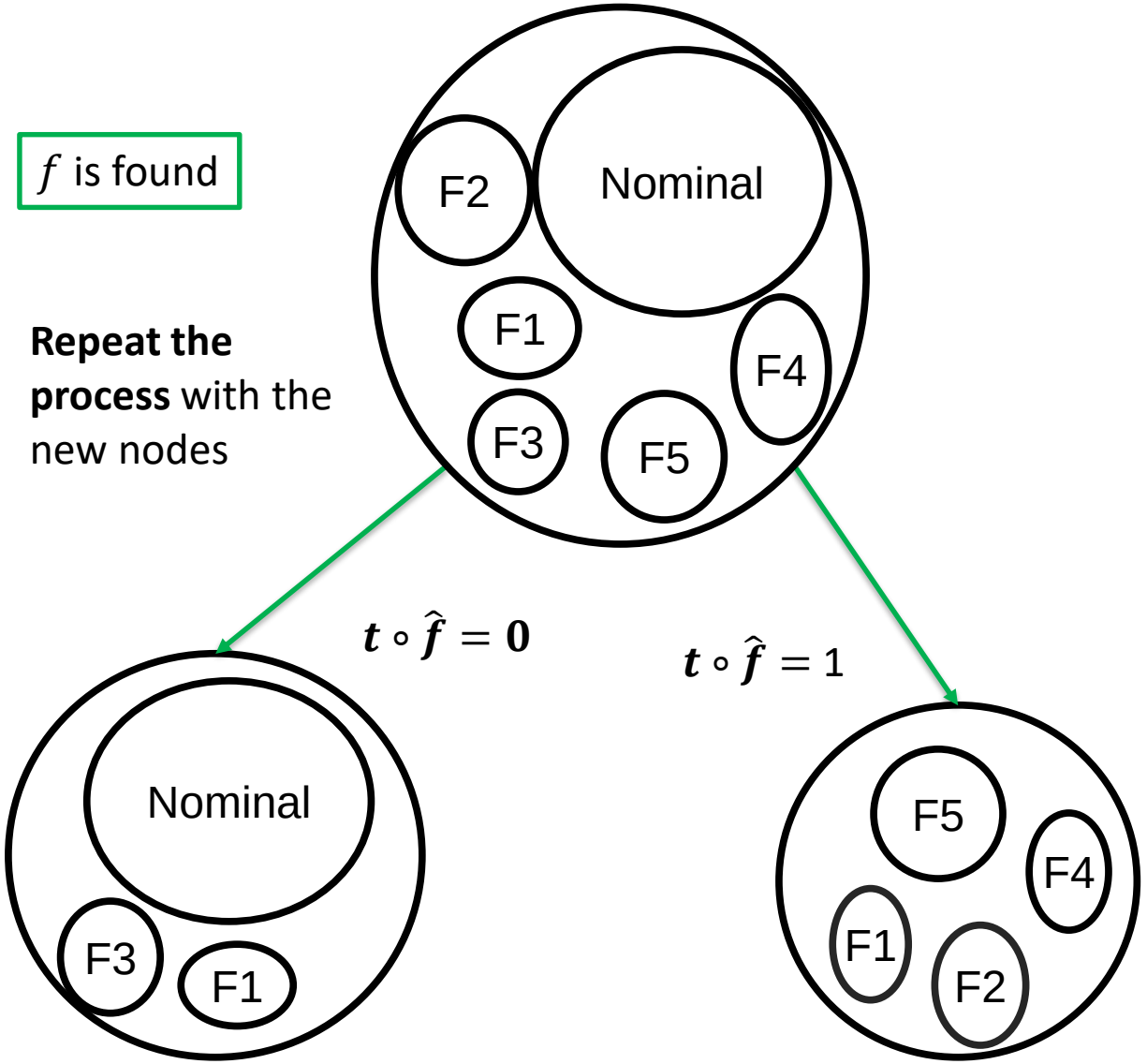
Repeat until you find f



DT4X - Principle

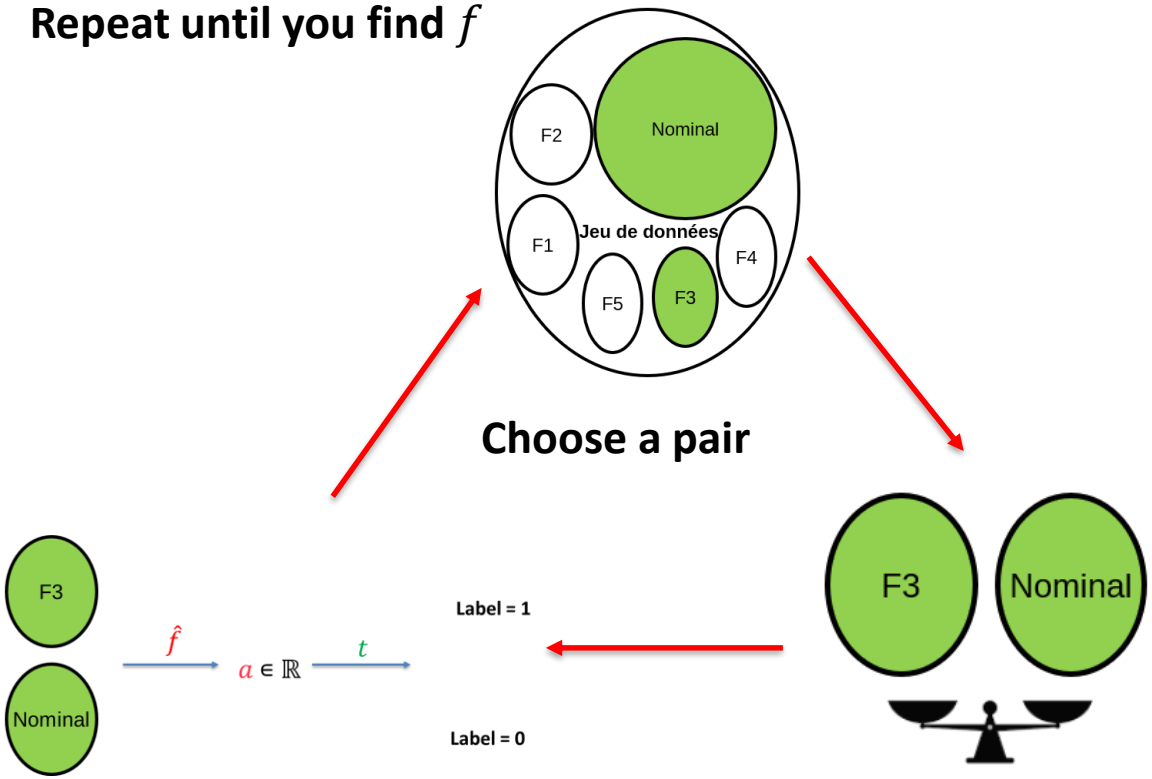
f is found

Repeat the process with the new nodes

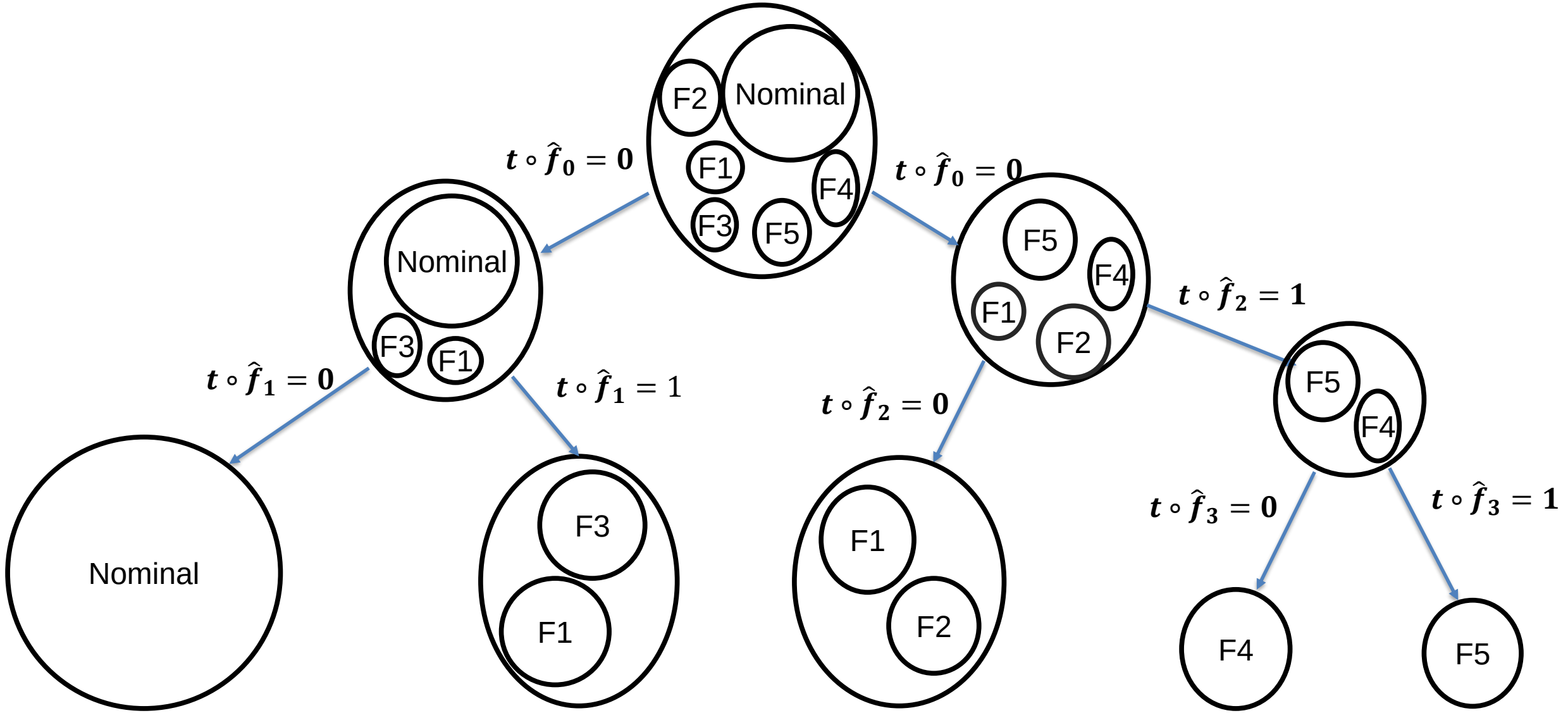


f is not found

Repeat until you find f



DT4X - Principle



Data-based ideal properties

Let $D = \{(x^i, \lambda^i)\}$ a dataset of labeled samples with

- $x^i \in \mathbb{R}^n$ a vector of observable variables
- $\lambda^i \in \Lambda = \{\lambda_k\}_{k=0\ldots n_F}$ the set of labels with λ_0 the nominal label and $\lambda_1, \ldots, \lambda_{N_F}$ the faulty labels associated to faults $F_1, \ldots F_{n_F}$

Data-based ARR

A relation $\mathrm{R}(\dot{x}', \ddot{x}', \ldots) = \mathrm{r(t)}$, with input x', a subvector of the observables x and output a scalar r, named residual, is a **data-based ARR** for the dataset D if, for all x such that $(x, \lambda_0) \in D$, it holds that $r =_\epsilon 0$.

Data-based ideal properties

Let C be a set of classes

Based on observable variables obs, we partition D so that there exists a surjective application $\delta: \mathbb{R}^n \rightarrow C$ such that $\delta(x^i) = C^i$

A fault F_k associated to label λ_k is **data-based detectable** on a dataset D if and only if there exists a surjective application $\delta: \mathbb{R}^n \rightarrow C$ such that

$$\forall\ (x, \lambda_k) \in D, k \neq 0, \forall\ (y, \lambda_0) \in\ D, \delta(x) \neq \delta(y)$$

→ A fault F_k is said to be **data-based detectable** on a dataset D if there exists a surjective application that matches faulty samples labeled λ_k and nominal samples labeled λ_0 to different outputs. Otherwise, it is **non-detectable**.

Two faults F_j and F_k associated to labels λ_j and λ_k are **data-based isolable** on a dataset D if and only if there exists a surjective application $\delta: \mathbb{R}^n \rightarrow C$ such that

$$\forall\ (x, \lambda_j) \in D, \forall\ (y, \lambda_k) \in\ D, \delta(x) \neq \delta(y)\ (j \neq k)$$

→ Two faults are data-based isolable if there exists a surjective application that matches faulty samples labeled by the 2 different faults to different outputs. Otherwise, they are **non-isolable**

Data-based diagnosability on D: data-based diagnosability encompasses both data-based detectability and data-based isolability.

Data-based practical properties

A relation $\mathrm{R}(\dot{x}', \ddot{x}', \dots) = \mathrm{r(t)}$, with input x', a subvector of the observables x, and output a scalar r, named residual, is a **data-based ARR** for the dataset D if, for **at least a proportion $\alpha \in]0,1], \alpha \approx 1$ of samples x** ~~for all x~~ such that $(x, \lambda_0) \in D$, it holds that $r =_\epsilon 0$.

A fault F_k associated to label λ_k is **data-based detectable** on a dataset D if and only if there exists a surjective application $\delta: \mathbb{R}^n \to C$ such that **at least a proportion $\alpha \in]0,1], \alpha \approx 1$ of samples x and of samples are such that**

$$(x, \lambda_k) \in D, k \neq 0, (y, \lambda_0) \in D, \qquad \delta(x) \neq \delta(y)$$

Two faults F_j and F_k associated to labels λ_j and λ_k are **data-based isolable** on a dataset D if and only if there exists a surjective application $\delta: \mathbb{R}^n \to C$ such that **at least a proportion $\alpha \in]0,1], \alpha \approx 1$ of samples x and of samples y are such that**

$$(x, \lambda_j) \in D, (y, \lambda_k) \in D, \quad \delta(x) \neq \delta(y)\ (j \neq k)$$

Data-based diagnosability on D: data-based diagnosability encompasses both data-based detectability and data-based isolability.

Data-based diagnosability deduced from DT4X tree

Let T be the decision tree produced by DT4X, and $L(T)$ the set of its leaf nodes
$D_L = \{(x, \lambda) \in D, x \text{ reached leaf } L\}$ the subset of samples assigned to leaf L
$\mathcal{L}_L = \{\lambda, (x, \lambda) \in D_L\}$ is the set of labels in leaf L

A fault F_k associated to label λ_k is **data-based detectable** iif:

$$\forall\, L \in L(T), \lambda_0 \in \mathcal{L}_L \Rightarrow \lambda_k \notin \mathcal{L}_L$$

→ A fault F_k is said to be **data-based detectable** on a dataset D if no leaf contains both nominal and λ_k labels
Otherwise, it is **non-detectable**.

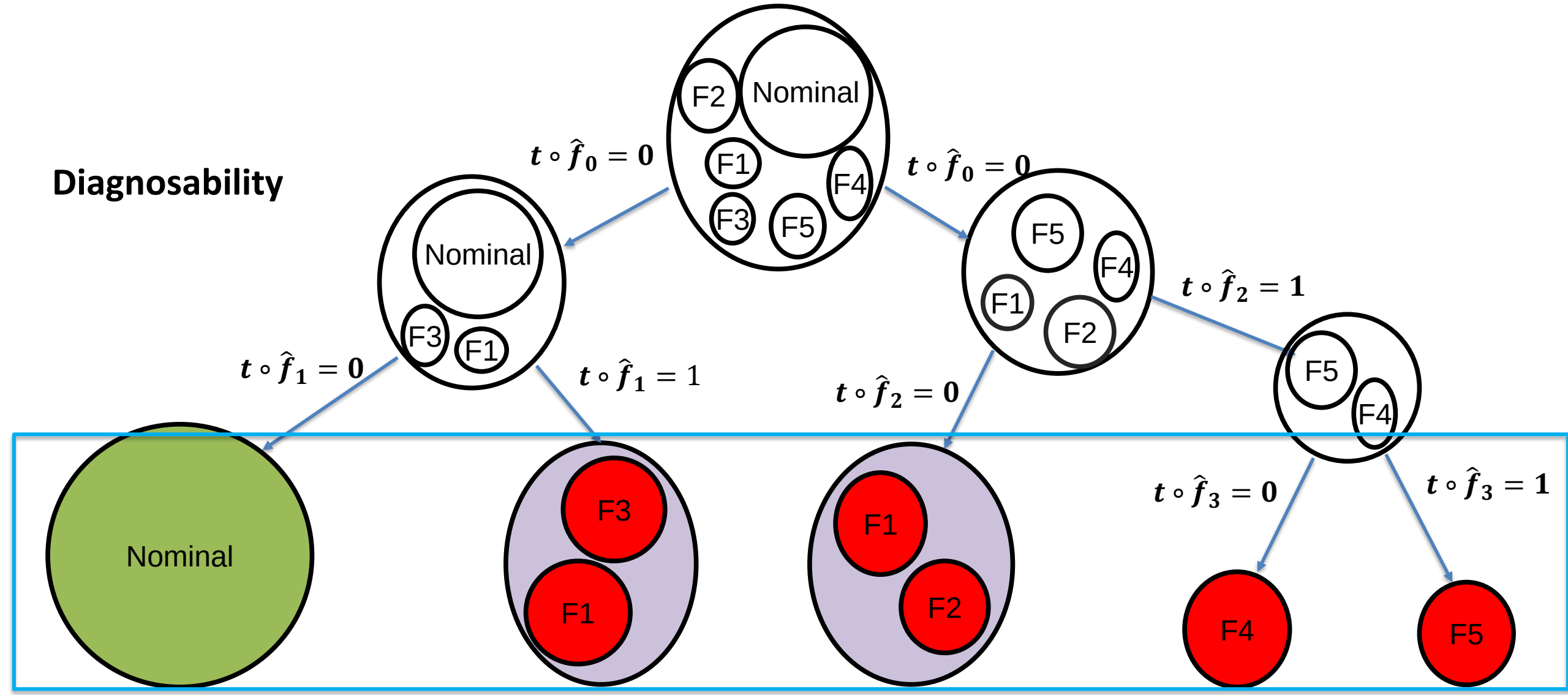
Two faults $\mathrm{F_j}$ and F_k associated to labels λ_j and λ_k are **data-based isolable** iif:

$$\forall\, L \in L(T), \lambda_i \in\ \mathcal{L}_L \Rightarrow \lambda_k \notin\ \mathcal{L}_L$$

→ Faults $\mathrm{F_j}$ and F_k are **data-based isolable** if their labels never appear together in the same leaf. Otherwise, they are **non-isolable**.

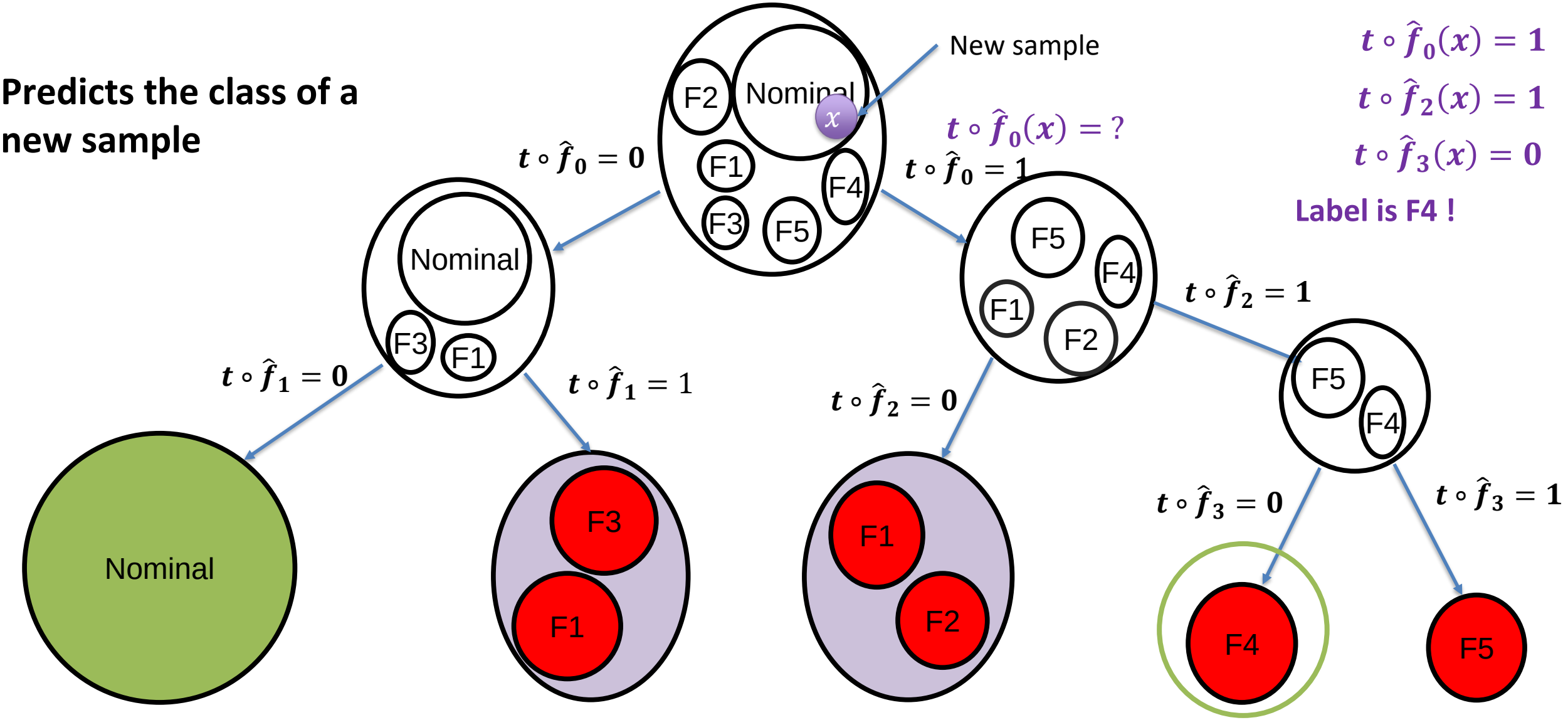
- ❑ Data-based detectability and isolability are induced by the structure of the DT4X tree
- ❑ They depend on how data samples are grouped into leaves that correspond to distinct behavioral relations $\mathrm{R}(obs) =_\epsilon 0$
- ❑ Diagnosability is not infer from a model but directly from data through the DT4X tree → data-based diagnosibility

DT4X - Principle



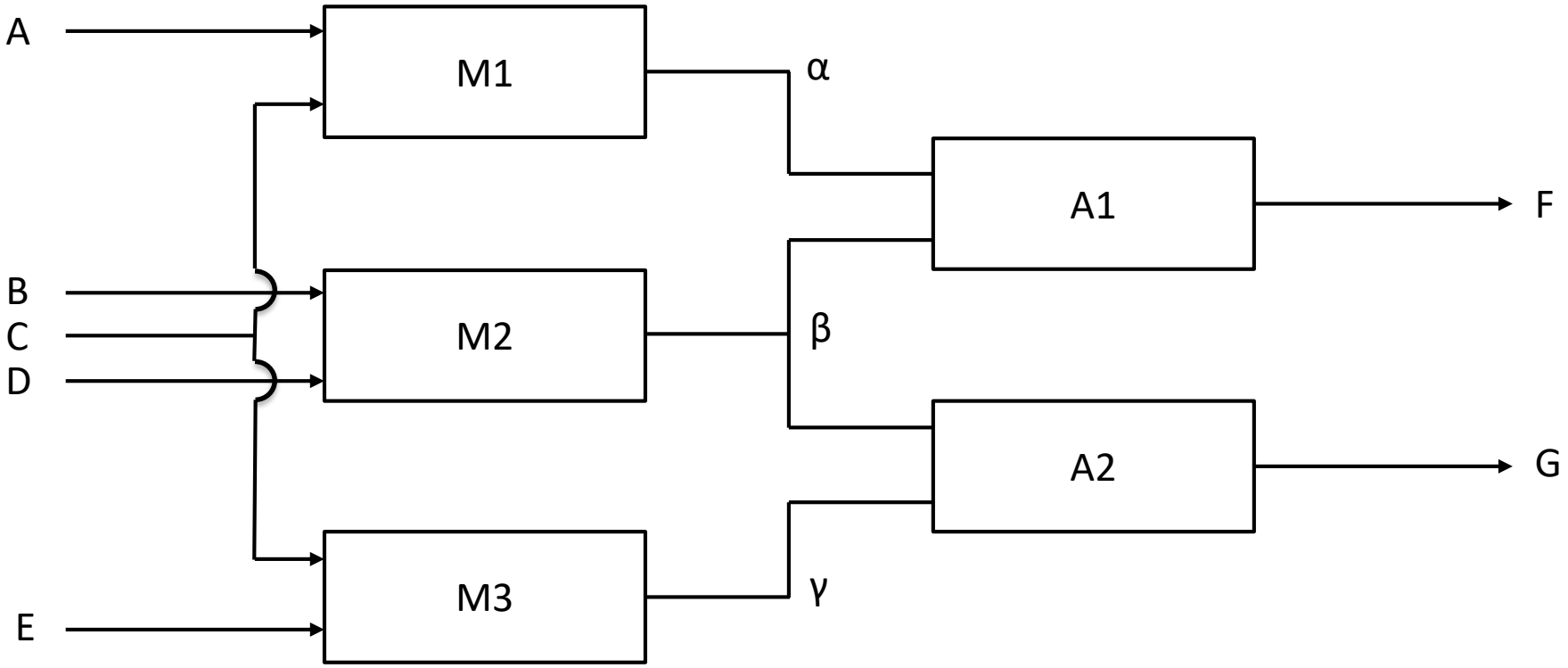
DT4X - Principle

Predicts the class of a new sample



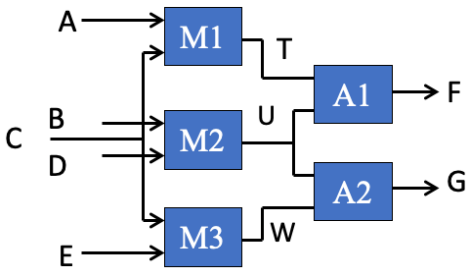
Polybox

Type: static
Known ARRs: yes
Data: simulated



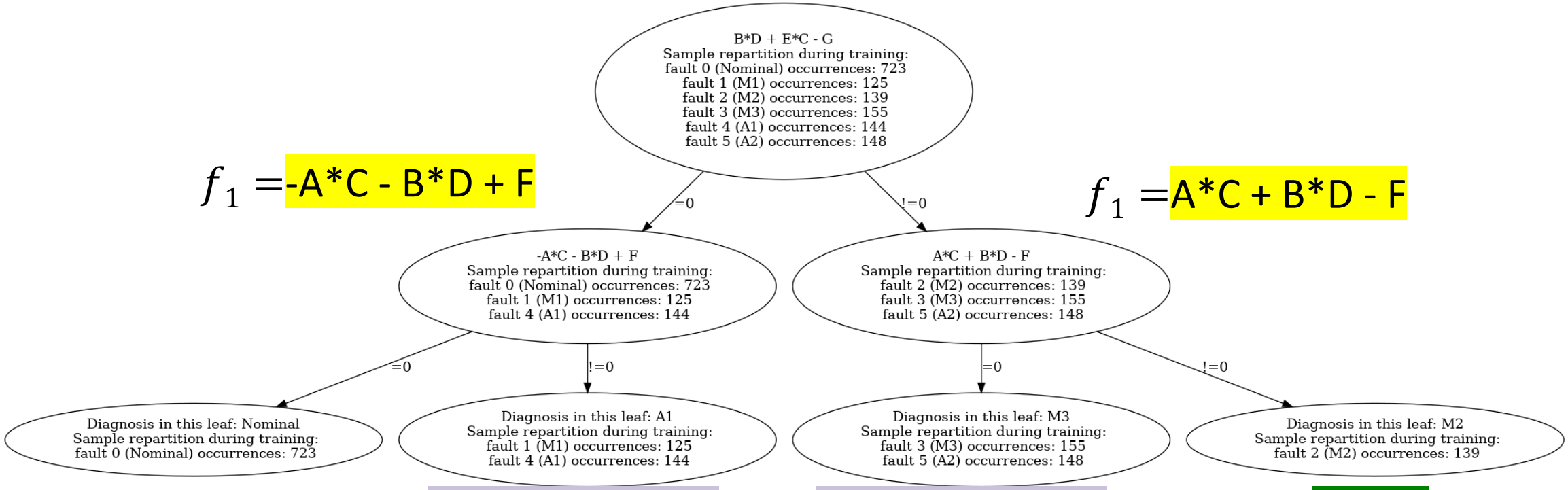
Louis Goupil, Elodie Chanthery, et al. (2023). "Tree based diagnosis enhanced with meta knowledge". In: 34th International Workshop on Principles of Diagnosis (DX'23).

PolyBox Results – Single Faults



	ok	Faulty M1	Faulty M2	Faulty M3	Faulty A1	Faulty A2
R1 : A*C + B*D – F = 0	0	X	X	0	X	0
R2 : E*C + B.D – G = 0	0	0	X	X	0	X
R3 : A*C – E*C – F + G = 0	0	X	0	X	X	X

$$f_0 = B*D + E*C - G$$



$$f_1 = -A*C - B*D + F$$

$$f_1 = A*C + B*D - F$$

Nominal

Fault M1 & Fault A1

Fault M3 & Fault A2

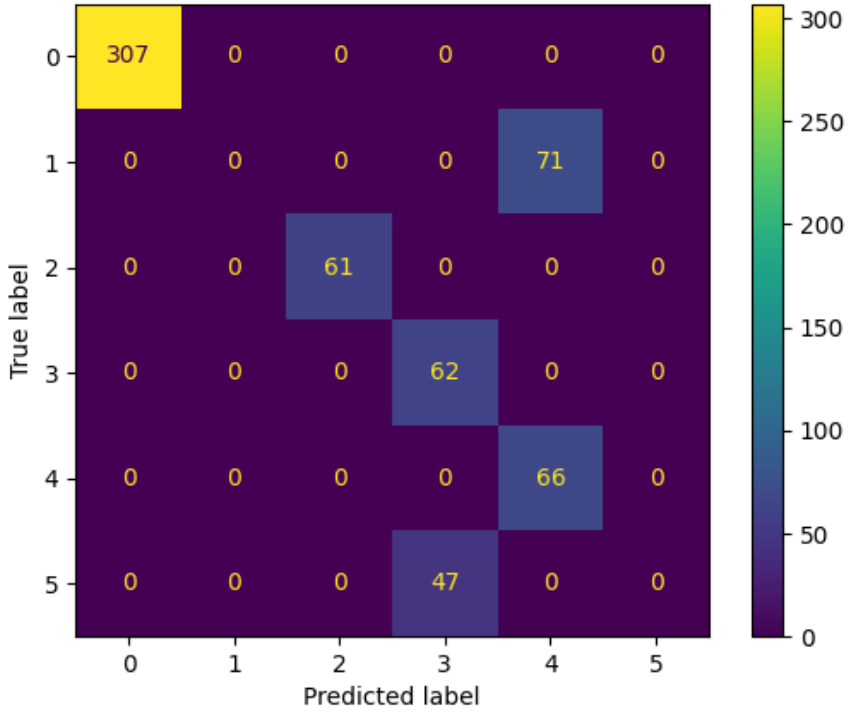
Fault M2

Polybox - Single faults

$$f1 - score = \frac{2TP}{2TP + FP + FN}$$

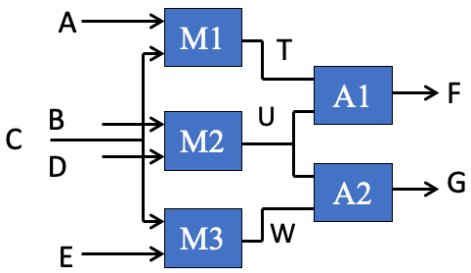
$$Accuracy = \frac{TP + TN}{P + N}$$

Method	Accuracy	F1 - scores	Inference Time	Optimal minimal tree depth	Extra parameters
HHCART	0.49052	0.35725	0.18s	20	
CO2	0.51045	0.36537	0.09s	4	
Oblique L-BFGS	0.53938	0.47204	0.002s	25	
OC1	0.49534	0.33698	2.78s	15	
RandomForest	0.41241	0.33375	0.07s	Default	
LogisticRegression	0.49309	0.32568	0.003s	Default	
NaiveBayes	0.49309	0.32568	0.01s	Default	
SVC	0.49309	0.32568	2.66s	Default	
KNN_Classifier	0.47219	0.36238	0.04s	Default	
Stree	0.50787	0.36471	1.6066s	15	kernel='rbf'
XGBoost	0.57666	0.50772	0.02s	6	n_estimators=200
SREDT	0.79781	0.77761	0.0009s	5	max_expression_hei function_set= ('mul', 'add', 'sub')
DT4X *	0.79909	0.73210	0.00074s	3	



* DT4X also finds the analytical expression of model based ARRs

PolyBox Results – Double Faults



$f_0 = -B*D - E*C + G$

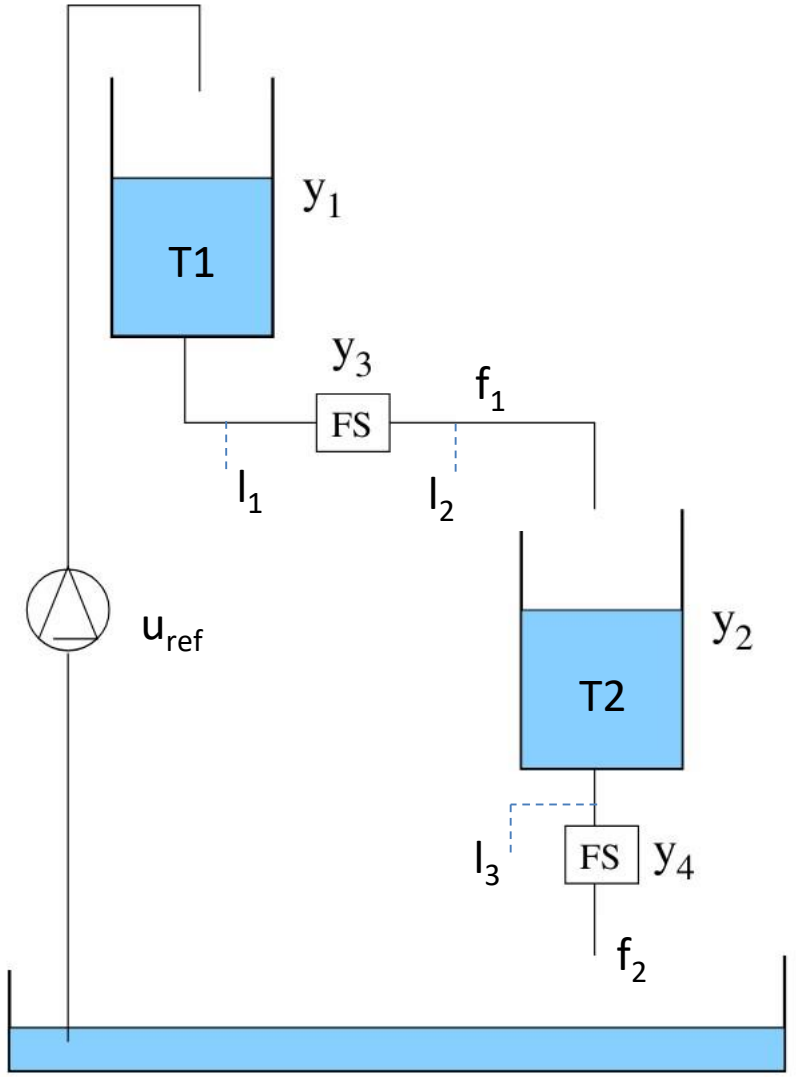
	ok	Faulty M1	Faulty M2	Faulty M3	Faulty A1	Faulty A2
R1 : A*C + B*D – F = 0	0	X	X	0	X	0
R2 : E*C + B.D – G = 0	0	0	X	X	0	X
R3 : A*C – E*C – F + G = 0	0	X	0	X	X	X

Dynamic systems

Static: $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$

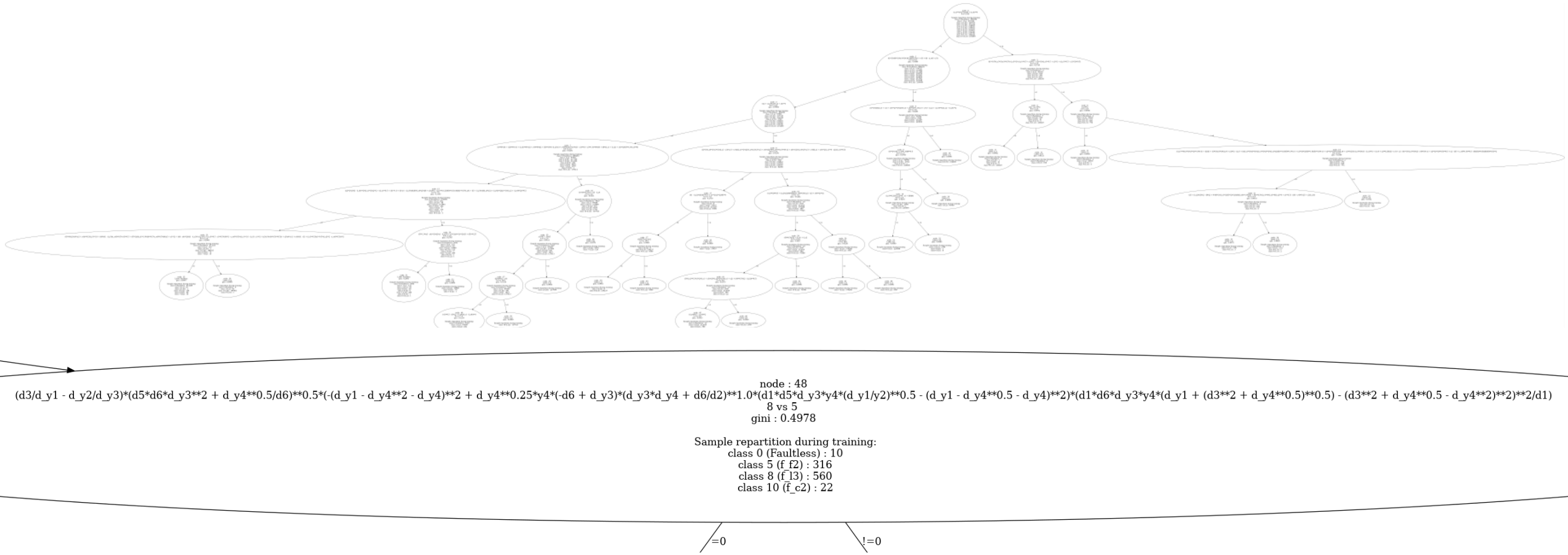
Dynamics: $x = (x_1, \dots, x_n, \dot{x}_1, \dots, \dot{x}_n, \dots, {x_1}^{(d)}, \dots, {x_n}^{(d)}) \in \mathbb{R}^{(d+1)n}$

Water tanks



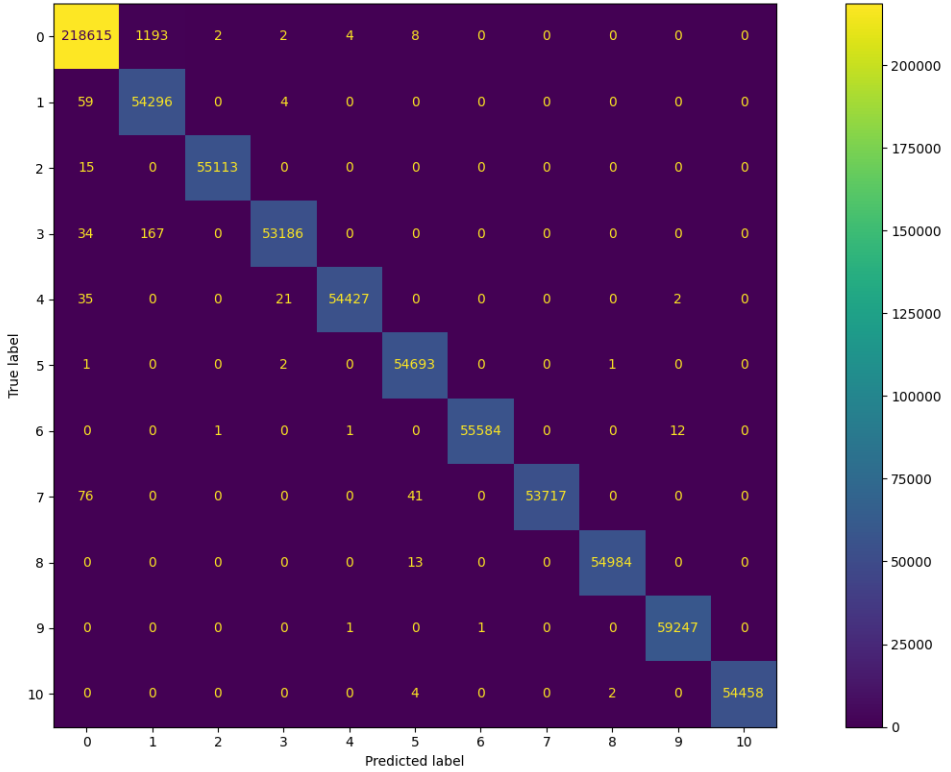
Type: dynamic
Known ARRs: yes
Data: simulated

Water tanks


node : 48
(d3/d_y1 - d_y2/d_y3)*(d5*d6*d_y3**2 + d_y4**0.5/d6)**0.5*(-(d_y1 - d_y4**2 - d_y4)**2 + d_y4**0.25*y4*(-d6 + d_y3)*(d_y3*d_y4 + d6/d2)**1.0*(d1*d5*d_y3*y4*(d_y1/y2)**0.5 - (d_y1 - d_y4**0.5 - d_y4)**2)*(d1*d6*d_y3*y4*(d_y1 + (d3**2 + d_y4**0.5)**0.5) - (d3**2 + d_y4**0.5 - d_y4**2)**2)**2/d1)
8 vs 5
gini : 0.4978
Sample repartition during training:
class 0 (Faultless) : 10
class 5 (f_f2) : 316
class 8 (f_l3) : 560
class 10 (f_c2) : 22
=0
!=0

Water tanks

Method	Accuracy	F1 - scores	Inference Time	Optimal minimal tree depth	Extra parameters
HHCART	0.32261	0.17241	0.98	4	
CO2	0.29381	0.133443	0.09	9	
Oblique L-BFGS	0.99954	0.99954	0.02	15	
OC1	0.61184	0.54916	15.9	30	
RandomForest	0.99979	0.99979	0.92s	Default	
LogisticRegression	0.72886	0.69385	0.02s	Default	
NaiveBayes	0.68449	0.67732	0.05s	Default	
SVC	0.92699	0.92513	1694.93s	Default	
KNN_Classifier	0.99900	0.99900	0.64s	Default	
Stree	0.99960	99960	142.7398s	14	kernel='linear' normalize=True
XGBoost	0.99973	0.99972	0.13s	Default	
SREDT	0.99768	0.99768	0.0733s	9	max_expression_height=4 function_set= ('mul', 'add', 'sub', 'sqrt')
DT4X	0.98604	0.98601	0.014s	5	



Please note:

- DT4X does not find model-based ARRs (syntactically) but data-based ARRs
- Isolation of classes structurally non-isolable
- Performance close to the best algorithms

Conclusions

- **DT4X** is not only a data-based diagnosis system but it discovers the expressions of diagnosis indicators that provide explainability
- **Paradigm shift**: identify input-output relations instead of finding separators
- **Message**: do not forget the interpretable concepts and results of model-based methods !

Ideas for future work

- **Boost symbolic classification** with information extracted from the physical model of the system to reduce computation time and improve reliability (mostly for dynamic systems)
- Compare the genetic algorithm with **deep symbolic regression (done)** or **reinforcement learning**
- Analyse fault non-diagnosability in relation to **symbolic classification convergence** (not good f)

L. Goupil, L. Travé-Massuyès, E. Chanthery, T. Kohler, and S. Delautier. Tree based diagnosis enhanced with meta knowledge applied to dynamic systems. *Best Theory Paul M. Frank paper Award,* IFAC Safeprocess , June 2024, Ferrara, Italy. *IFAC-PapersOnLine*, 58(4):1–6, 2024.