



Universiteit
Leiden
The Netherlands

Exploring hysteron design space using interpretable Machine Learning

Bontenbal, Paco

Citation

Bontenbal, P. (2024). *Exploring hysteron design space using interpretable Machine Learning*.

Version: Not Applicable (or Unknown)

License: [License to inclusion and publication of a Bachelor or Master Thesis, 2023](#)

Downloaded from: <https://hdl.handle.net/1887/3731924>

Note: To cite this publication please use the final published version (if applicable).



Exploring hysteron design space using interpretable Machine Learning

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
PHYSICS

Author :	P. Bontenbal
Student ID :	s3746992
Supervisor :	prof. dr. M.L. van Hecke
Daily supervisor :	Ryan van Mastrigt
Second corrector :	prof. dr. K.E. Schalm

Leiden, The Netherlands, March 15, 2024

Exploring hysteron design space using interpretable Machine Learning

P. Bontenbal

Huygens-Kamerlingh Onnes Laboratory, Leiden University
P.O. Box 9500, 2300 RA Leiden, The Netherlands

March 15, 2024

Abstract

In the pursuit of designing complex materials with desired properties, understanding their design parameter space is crucial. However, this space's convolution often hinders comprehension of complex materials' responses as a function of their design parameters. Machine Learning has recently emerged as a promising tool for capturing patterns in complex design spaces, although this performance often comes at the cost of interpretability. This thesis aims to explore the design parameter space of interacting hysterons using interpretable Machine Learning, specifically Decision Tree inspired methods. Despite the complexity of the design parameter space of even small systems of interacting hysterons, interpretable Machine Learning can classify coarse-grained properties of the system effectively. Introducing the Support Vector Classifier (SVC) inspired Decision Tree, we achieve almost perfect isolation of these properties. This model preserves interpretability while effectively probing the statistical structure of design parameter space of systems of interacting hysterons.

Contents

1	Introduction	7
2	The model of interacting hysterons	9
2.1	Preisachs model of (non-interacting) hysterons	9
2.2	Model of interacting hysterons	10
2.3	Transition graphs	11
2.3.1	t-graph properties	12
3	Sampling the design parameter space	15
4	Probing the design parameter space	17
4.1	Decision trees	17
4.1.1	DT Results	19
4.2	Design Inequalities	20
4.3	Support Vector Machines	21
5	A Support Vector Classifier inspired Decision Tree	25
6	Results	27
6.1	2 hysterons	27
6.2	3 hysterons	29
6.3	Decisions	30
7	Conclusions and future prospects	31
7.1	Conclusion	31
7.2	Discussion	32
7.3	Future Prospects	32

Introduction

Many complex materials, from amorphous matter to glasses and meta-materials, are comprised of many meta-stable states [1–3]. These states can be adequately modelled by a system of interacting hysterons, hysteric two state elements [4, 5]. External driving can change the system from one state to another, creating pathways between these states. The collection of pathways between these states, represented by transition graphs, or t-graphs, show interesting properties, such as breaking return point memory [6], sub-harmonic orbits [5], and transient memory [7]. A system of n_h interacting hysterons has a finite set of possible t-graphs. This set of t-graphs grows exponentially with the number of interacting hysterons, making it difficult to fully enumerate the design parameter space for many interacting hysterons. This is why we are endeavoring to find rules that describe the design parameter space.

Previous work in this field [5, 8] has shown that it is possible to describe each possible transition graph of a system of 2 or 3 interacting hysterons. A transition graph, or t-graph is a visual representation of all reachable states of a system of interacting hysterons. [5] has already given us a set of pairwise inequalities, able to describe every possible t-graph in a system of 2 interacting hysterons. Recently, this work has been extended to creating precise and finite algorithms to obtain these inequalities for a system of an arbitrary number of interacting hysterons [8]. We expect sets of inequalities describing a specific t-graph to form a polytope in the design parameter space. As the presence of specific features, such as the presence of avalanches, can be easily decided for each t-graph, the subset in design

space associated with a common coarse-grained property is, in general, a collection of such polytopes. Yet, despite this framework of inequalities, we still lack understanding of these polytopes.

We have some, but not much, intuition about the frequency, location, and general shape of these polytopes in the design space. We would like to explore this further, especially for systems with more than 2 interacting hysterons. Even though complete sets of inequalities are known for systems of 2, and can be derived for systems of 3 interacting hysterons, we would like to have a better idea of the statistical structure of their design parameter spaces. Machine Learning has recently shown to be effective in finding rules that delineate convoluted high dimensional spaces, in design metamaterials [9–11] for example.

This thesis proposes an interpretable Machine Learning model, the Support Vector Classifier (SVC) inspired Decision Tree (DT) to identify coarse-grained sub-spaces. These sub-spaces are defined by common properties of t-graphs that reside in these sub-spaces. We first try to do so using Decision Trees (DTs) (Chap. 4.1), but find their decision boundaries to be lacking in expressiveness as they are only 1st order inequalities. We therefore introduce the SVC inspired Decision Tree DT (Chap. 5), that preserves DT structure, but incorporates a more expressive decision boundary, in the form of a linear inequality of order d , where d is the number of design parameters. We find that this model, while retaining interpretability, is able to almost perfectly isolate these coarse grained regions defined by common properties, such as avalanches, for small systems of interacting hysterons (Chap. 6). We expect the SVC inspired DT to be a useful tool for probing the statistical structure of design parameter spaces of systems of interacting hysterons.

The model of interacting hysterons

In this thesis we will be looking at the design parameter space of systems of linearly interacting hysterons, as described in [5]. Therefore, the following section is dedicated to explaining this system and both a visualisation of it, in the form of transition graphs, as well as properties that these systems can exhibit. We will do so by first going over the Preisach model of (non-interacting) hysterons [4], after which we will introduce a linear interacting between these hysterons. Furthermore, it will introduce the transition graph as a means of visualising states and transitions in the system. Finally, it highlights some of the properties a system of interacting hysterons can have and elaborates on them.

2.1 Preisachs model of (non-interacting) hysterons

If we want to model metastable materials, such as programmable metamaterials and amorphous solids, we can approach this metastability as a consequence of many (interacting) bistable elements. Originally derived for ferromagnetism, a hysteron is a hysteric two state element that can either be 0 or 1 and allows us to model meta-stable states. Hysterons are based on the principle of hysteresis. Now hysteresis can mean many things depending on the context it is described in. However, in Preisach's model of hysteresis, hysteresis means that switching up, from state 0 to 1 and down, from 1 to 0, happens at two different values of an external driving field U . We call these values the switching fields U_i^+ and U_i^- respectively, for which

$$0 < U_i^{+,-} < 1. \quad (2.1)$$

Transitions in the system, from state $S \rightarrow S'$, where

$$S = \{s_1, s_2, \dots, s_{N_h}\} \quad (2.2)$$

describes a system of N_h hysterons, are a result of external driving of the system, where transitions take place when the external driving passes some critical value U_c . A hysteron i in state $s_i = 0(1)$ flips up(down) to state $s_i = 1(0)$ when the external driving field is larger(smaller) than the bare up(down) switching field; $U > U_i^+(U < U_i^-)$. The order in which hysterons switch up or down under driving is independent of the state of the system.

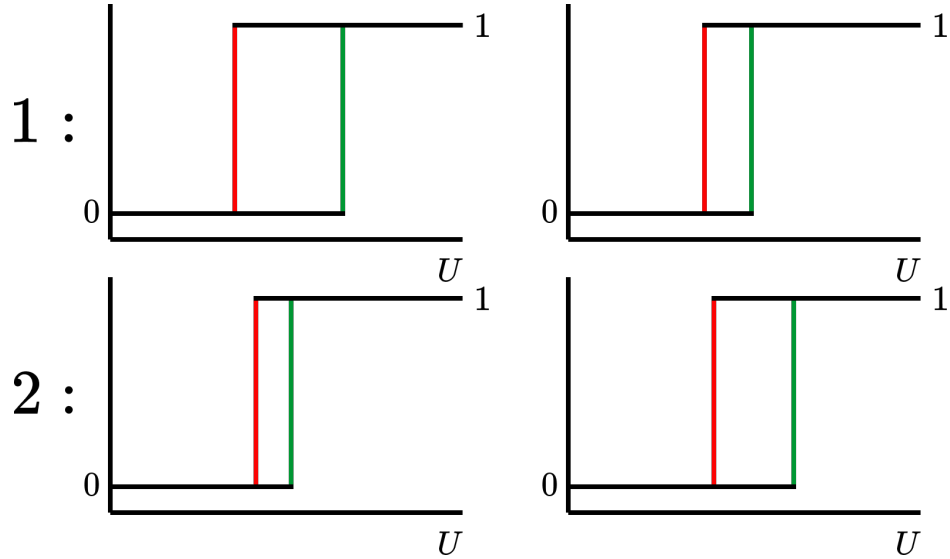


Figure 2.1: The switching fields can change as the systems transitions from state S (left) to a new state S' (right). The values of the down (red) and up (green) switching fields changes as a function of the state of the system, resulting in a change in the ordering of the switching fields.

2.2 Model of interacting hysterons

If we add linear interactions between these hysterons such that the switching fields $U_i^{+,-}$ ($0 < U_i^{+,-} < 1$) are now a function of the bare switching fields $u_i^{+,-}$ ($0 < u_i^{+,-} < 1$) and the interaction coefficients c_{ij} ($-1 < c_{ij} < 1$);

$$U_i^{+,-}(S) = u_i^{+,-} - \sum_{j \neq i} c_{ij} s_j, \quad (2.3)$$

we find that the switching fields of the hysterons are dependent on the state S of the system. Here we impose $u_1^+ > u_2^+ > \dots$ and assume no two bare switching fields to ever have the same value. This means that the ordering of the switching fields is also state dependent; going from state S to S' , switches the order of U_i^+ and U_j^+ (Fig. 2.1).

At each state the system has two critical values that follow from the switching fields of the individual hysterons. Namely, one up and one down critical switching field:

$$U_c^+(S) = \min_i U_i^+(S), \quad (2.4)$$

$$U_c^-(S) = \min_i U_i^-(S), \quad (2.5)$$

which states that the lowest (highest) up (down) switching field of any hysteron in state 0 (1) is the critical up (down) switching field of the system in state S [5]. This means that if $U > U_c^+$ or if $U < U_c^-$ the system becomes unstable evoking a transition ($S \rightarrow S'$). For a more detailed description of the linearly interacting hysteron model, see [5, 8].

2.3 Transition graphs

Transitions graphs, or t-graphs, introduced in [5], are a visualisation of all reachable states and possible transitions between these states of a system of hysterons. In order to generate a transition graph, we go through all the states the system can reach from the two extremal states, $\mathbf{0} = \{0, \dots\}$ and $\mathbf{1} = \{1, \dots\}$. The states are then displayed in the t-graph in order of low to high magnetisation from bottom to top, where $m = \sum_i s_i$. Furthermore, the states are arranged lexicographically from left to right. We see in Figure 2.2 all possible t-graphs for a system of 2 linearly interacting hysterons. This set includes the two graphs (in green) that correspond with the Preisach model of non-interacting hysterons. The addition of interaction, makes a lot more t-graphs realisable (in orange). We have two bare switching fields for each hysterons, giving us $2n_h$ design parameters. Additionally, considering interaction coefficients c_{ij} where $c_{ii} = 0$, there are $n_h^2 - n_h$ such coefficients. The number of design parameters for a system of hysterons is therefore $n_h^2 + n_h$. The relationship between the number of interacting hysterons and the number of realisable t-graphs is exponential. It grows from 2 to 11 for a system of 2 hysterons and we know that for a system of 3 hysterons the possible t-graphs grow from 6 to over 15000.

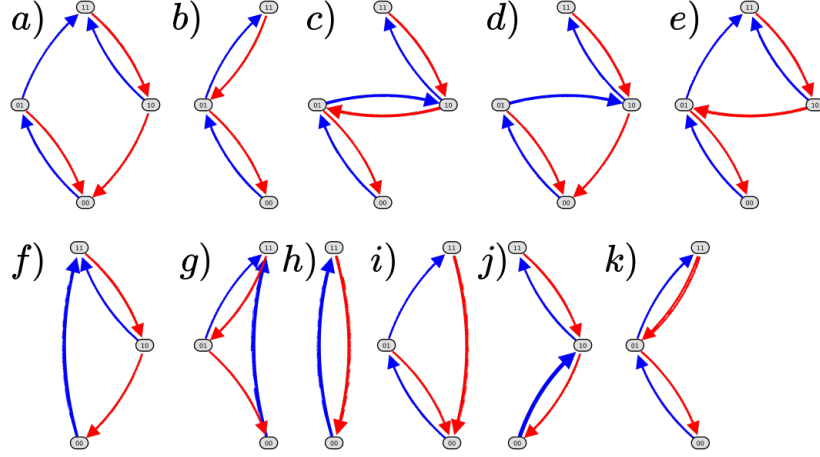


Figure 2.2: All possible transition graphs for a system of 2 interacting hysterons. We find the Preisach t-graphs, where $c_{ij} = 0$ (a, b). As well as the t-graphs that additionally become possible when we add linear interactions $c_{ij} \neq 0$ between hysterons (c-k). This Figure is a modification of the original from [5]

2.3.1 t-graph properties

Figure 2.2 shows that new transitions arise when we introduce linear interactions between hysterons. Most notably, avalanches, Figure 2.2 (c-k), are a new transition graph property. Another important new phenomenon is that transition graphs can now be ‘ill defined’. We require this new definition as it is possible for transitions to end in self loops, $S \rightarrow S$.

In order to understand how these properties come to be, let's look at a case where we transition from state $S = \{00\}$ for a system with $n_h = 2$. As mentioned in the previous section, if the external driving field is bigger than the critical up switching field of state $\{00\}$ (2.4), the system will transition to a landing state S' . Since $u_2^+ < u_1^+$ is imposed, the system will transition to $S' = \{01\}$ at $U_c = u_2^+$. It is not a given, however that this state is stable. $S' = \{01\}$ has two critical switching fields itself, see Figure 2.3. The critical driving field $U_c = u_2^+$ must therefore be greater (lesser) than the critical down (up) switching field of the landing state $S' = 01$ for S' to be a stable landing state. This means $S' = \{01\}$ is unstable if $u_2^+ < u_2^-$, which is a given, or if $u_2^+ > u_1^+ - c_{12}$. Generally, any landing state S' is unstable if the driving field U is greater (lesser) than the critical down (up) switching field of S' . If $S' = \{01\}$ is unstable, that means that a

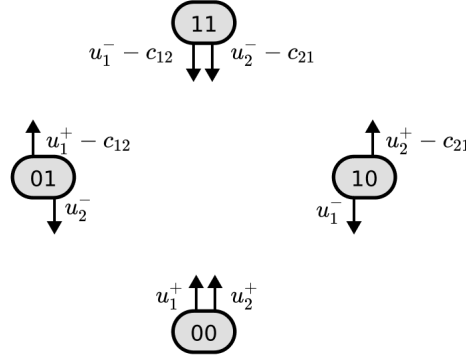


Figure 2.3: Shown here are the critical switching fields of all possible states of a system of 2 hysterons. The arrows indicate the possible transitions of each hysteron. Figure from [5]

transition from S to $S'' = \{11\}$ is possible. We follow the same procedure as before and again find condition(s) in the form of pairwise inequalities for this transition to be stable. If it is, we call this transition $S \rightarrow S''$ an 'avalanche' transition. Generally, any transition between two states where more than one hysteron switches up or down is called an 'avalanche' transition. A case in which a transition from a state ends up back at that same state $S \rightarrow S$ we call 'ill defined'.

Below we find a decision tree, which shows what transition is made, based on the pairwise inequality conditions that we found. Doing this for every

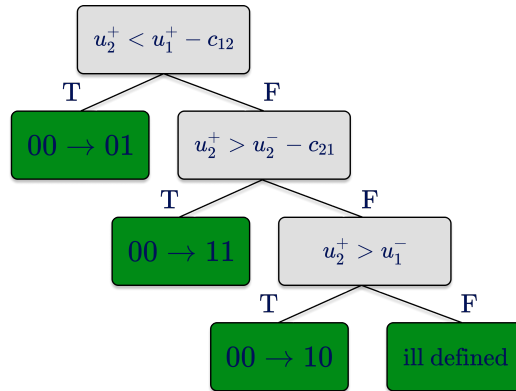


Figure 2.4: Decision tree of pairwise inequality conditions derived from the critical switching fields, that shows a transition from state 00. 'T' ('F') indicates a condition to be true (false). Figure from [5]

state that is reachable, meaning that a transition to that state is stable, com-

pletes a transition graph.

Some properties only arise in systems with a larger number of hysterons. One such property is the breaking of loop-Return Point Memory (l-RPM). We define a loop to have a 'bottom' state and a 'top' state and for these states to be connected by two sequences, one of up transitions, the other of down transitions. l-RPM requires for a sequence starting from any state in this loop to only be able to go to the 'bottom' or 'top' state. If these conditions are not met, l-RPM is broken. An example is shown in Figure 2.5. The sequence $\{100\} \rightarrow \{110\} \rightarrow \{111\}$ breaks l-RPM of the loop between 'bottom' state $\{000\}$ and 'top' state $\{101\}$. For a more elaborate explanation of t-graph creation and property definitions, such as ill definition, avalanche(s) and breaking l-RPM see [5, Section IIIB].

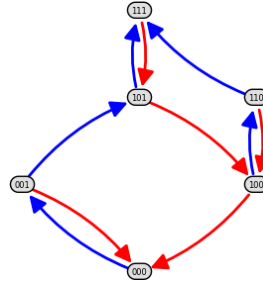


Figure 2.5: An example t-graph that breaks loop-Return Point Memory (l-RPM) for a system of 3 interacting hysterons. l-RPM is broken by the sequence $\{100\} \rightarrow \{110\} \rightarrow \{111\}$. Figure from [5].

Sampling the design parameter space

The essence of this thesis boils down to the following. In order to better understand the design space of interacting hysterons, we aimed to identify sub-spaces in that design parameter space that house t-graphs with a common property, a subspace where all t-graphs have one or more avalanches for example. In the following section we describe how we translate the properties from Sec. 2 into Machine Learning data, such that we can train a model to identify sub-spaces in the design parameter space based on common topologies of t-graphs in that sub-space. The transition graphs from the previous section are merely a nice visualisation of the relation between the design parameters $u_i^{+,-}$ and c_{ij} and the properties of the system, that we can label. Each data entry is therefore a d -dimensional point in the design parameter space, where $d = n_h^2 + n_h$ with assigned to it labels that are about the properties of the system. Recall that n_h is the number of interacting hysterons in the system.

$$x = \{u_i^{+,-}, c_{ij}\} \quad (3.1)$$

$$y = \{f\}, \quad (3.2)$$

where f is a statement about a property of the system, such as it being ill defined or having one or multiple avalanche(s).

These properties can be boolean statements, such as whether a t-graph is ill defined or not, although in some cases it is useful to save more information. Avalanches for example can occur more than once in a t-graph,

and there are multiple kinds of avalanches, that can happen between different states. Therefore we save information about avalanches as a list of strings of states between which avalanches occur. For example, the list for the graph in Fig. 2.2(c) is ;

$$\text{avalanches} = [0110, 1001]. \quad (3.3)$$

This allows us to also classify t-graphs with specific avalanche transitions instead of just whether t-graphs have avalanche(s) or not. Practically, we store boolean statements as 1 (true) or -1 (false), as we require a numeric value for our loss function to work.

We expect sub-spaces where all t-graphs have a common property to have the form of polytopes or a collection of polytopes. We do so based on the fact that it has been shown that a one or more pairwise inequalities can describe individual transitions, as well as complete t-graphs. In the design parameter space these pairwise inequalities form hyperplanes and we expect a collection of them to intersect to form a polytope. That is why, in order to find these polytopes, the following Section is dedicated to probing the design space with interpretable Machine Learning models that produce linear decision boundaries, that form hyperplanes in the design parameter space, recursively.

Probing the design parameter space

In this section, we introduce two models. The first is a Decision Tree (DT). We show how it performs on our data (Sec. 3) and try and explain the performance with a visual example from our data. We introduce the linear Support Vector Classifier (SVC) and show how it performs better on our example. Finally, we propose dressing the DT with a linear SVC, increasing its expressiveness, at each node. We call this model the Support Vector Classifier (SVC) inspired Decision Tree (DT).

4.1 Decision trees

A decision tree [12–14], one of the most famous and simple machine learning models, is a natural choice for our problem because transitions in our system (Sec. 2) follow a tree like structure (Fig. 2.4) as well. Another advantage of decision trees is that they have very interpretable results, as they offer a readable tree like structure of relatively simple questions about the data. Therefore, we use it for a first exploration of the design parameter space of t-graphs. Here, we will explain how a DT works.

A decision tree is a supervised learning model that can perform both regression and classification tasks. It starts with a node containing all data and tries to split the data into two parts, in such a way that it maximises the information gained about the data with respect to the classification task.

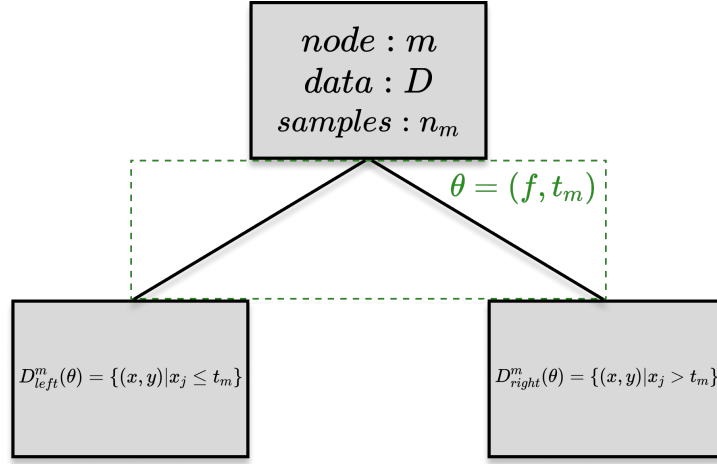


Figure 4.1: Representation of a single node in a decision tree, the green highlights the splitting of the data in node n according to $\theta = (f, t_m)$.

We can mathematically formulate the best split the following way. Say we have our data $D_m = (X, Y)$ at node m of n_m samples, and we want to make the best split and we do a split $\theta = (f, t_m)$, where f is a feature to split on and t_m , the threshold value of that feature, that splits the data into $(X, Y)_L^m$ and $(X, Y)_R^m$. We are then left with two subsets:

$$(X, Y)_L^m(\theta) = \{(x, y) | x_i \leq t_m\} \quad (4.1)$$

$$(X, Y)_R^m(\theta) = \{(x, y) | x_i > t_m\} \quad (4.2)$$

Then, the quality of this split with regards to the information gain is tested using a loss function. This is often either the Entropy or, as in this thesis the Gini impurity loss function

$$H(D_m) = \sum_k p_{mk}(1 - p_{mk}), \quad (4.3)$$

where p_{mk} is the ratio of class k observations in node m , defined as

$$p_{mk} = \frac{1}{n_m} \sum_{y \in D_m} I(y = k), \quad (4.4)$$

where $I(y = k)$ is the identity function if y is of class k . We use the Gini impurity loss function over the Entropy loss because it is generally computationally faster at no significant performance cost. We can then find

the best quality split by altering the split parameters $\theta = (f, t_m)$, such that it minimises:

$$G(D_m, \theta) = \frac{n_m^{left}}{n_m} H(D_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} H(D_m^{right}(\theta)). \quad (4.5)$$

To summarise; at each node m , a split $\theta = (f, t_m)$ that is parameterised by a feature and a corresponding threshold is chosen such that it minimises the impurity $G(D_m, \theta)$, see Figure 4.1. This means that at each split the model tries to gain the most information about the classification. The DT keeps splitting until all of the nodes are pure (or another termination condition is met), meaning that all of the training data samples in each node have the same label. We call those nodes leaf nodes.

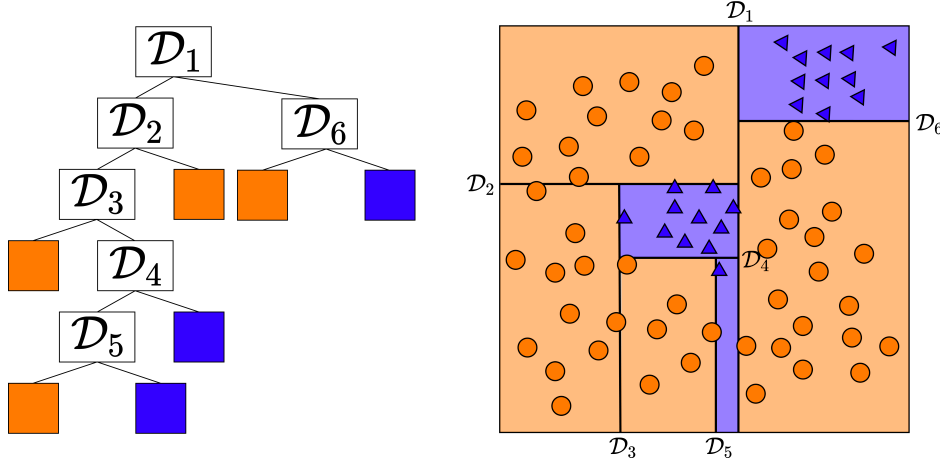


Figure 4.2: On the left we see an illustration of a DT that has been trained to solve the classification problem that is shown on the right, where the blue triangles and the orange circles represent two distinct classes. The colored regions on the right correspond with the colored leaf nodes on the left and indicate that the tree classifies all points in those regions as the class corresponding to that color.

Figure 4.2 shows the tendency of DTs to over-fit, since it can only make axis parallel decisions with a decision boundary. We also see that with increasing tree depth and thus increasing the number of leaf nodes results in more detailed decision regions.

4.1.1 DT Results

We show in this subsection the results of training DTs on the design parameters for a system of 2 interacting hysterons. In this section and the

ones after it, we will discuss two features. First, 'ill definition', because it is a simple boolean features for 2 hysteron. Furthermore, we will be looking at 'avalanches', both if they occur, and in what form they occur. We see

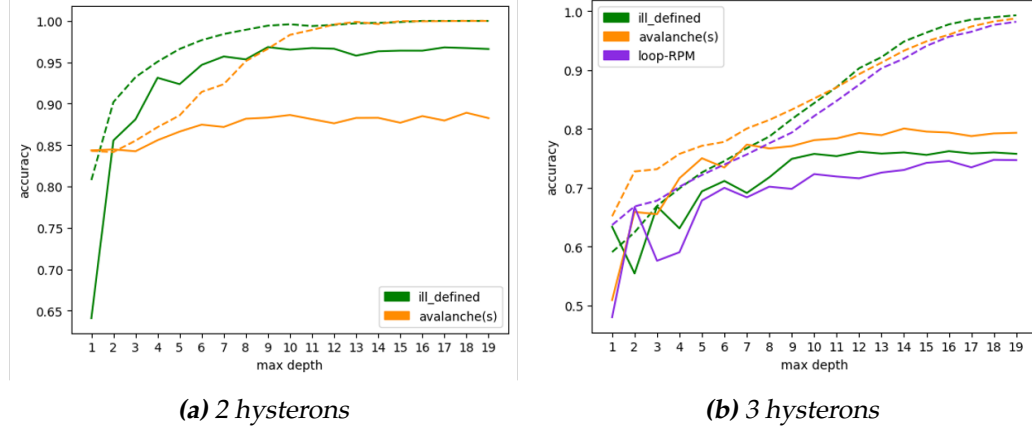


Figure 4.3: For a system of 2 and 3 interacting hysteron: Training- (dashed) and test (solid) accuracy of two (three) feature classifications, ill definition, avalanche(s) (and loop-RPM), as a function of the depth of the DT.

that the test accuracy for a system of 2 hysteron, Figure 4.3(a), converges at quite high, but sub-optimal accuracy of about 0.95 and 0.85 for 'ill definition' and 'avalanche(s)' respectively. We conjecture that the reason the former converges at higher accuracy than the latter is that there are many different types of avalanches, and thus many t-graphs that are classified as having avalanches. In the design parameter space that could mean that the sub-region that we are trying to classify might consist of multiple separate sub-regions, or clusters. Also, note the over-fitting of the DT, seen in the Figure as a separation between the training and test accuracies, indicative of bad generalisation, likely due to the DTs limitation to axis parallel decision making that only considers one design parameter.

4.2 Design Inequalities

[5] shows that for a system of 2 interacting hysteron it is possible to find a set of pairwise inequalities that is able to describe every possible transition graph. Furthermore, [8] shows that the same is possible for a system of 3 interacting hysteron. The set of pairwise inequalities for a system of 3

interactions is bigger than one for a system of 2 and it will grow with the number of interacting hystérons in the system. [5] also shows that a set of pairwise inequalities is able to describe all t-graphs that are 'ill defined' or have 'avalanche(s)'. If we train our DT with these pairwise inequalities x_i , found in [5] Table I, as the input data, it achieves an accuracy of 1, and also finds the set of conditions on a subset of pairwise inequalities that describe all t-graphs that are 'ill-defined'. x_i means that the condition is true while $\neg x_i$ means that the condition is not true. Interestingly, the DT finds a smaller subset than [5], namely:

$$\neg x_8 \wedge x_5 \wedge \neg x_6 \wedge \neg x_7, \quad (4.6)$$

instead of:

$$\neg x_3 \wedge x_5 \wedge \neg x_6 \wedge \neg x_7 \wedge \neg x_8. \quad (4.7)$$

This indicates but does not conclusively posit, that there is a potential redundancy in the set of inequalities presented in [5].

The performance of the DT on classification of properties of a system of 3 interacting hystérons has similar problems as described above. But these are amplified, we see even more over-fitting as well as lower test accuracy convergence. We attribute these issues to the higher dimensionality of both the parameter space and the clusters the DT is tasked to classify. We suspect that these clusters are more widely dispersed within the design space, exacerbating the difficulties faced by the DT. This situation underscores the limitations of employing single-axis parallel decision-making strategies in such complex systems.

To investigate this suspicion, we look at an example of non-axis parallel, but linearly separable data, we see why the performance of a DT might be sub optimal, see Figure 4.4. We see that in order to achieve decent test accuracy, the DT needs at least a depth of 3, with 6 leaf nodes (number of red dotted rectangles). We also see that in this case, the model is already over-fitting to the training data. For a better performance and to reduce over-fitting a more expressive decision boundary might be better.

4.3 Support Vector Machines

For a more expressive decision boundary we look to the linear Support Vector Classifier (SVC). Another very common, computationally efficient

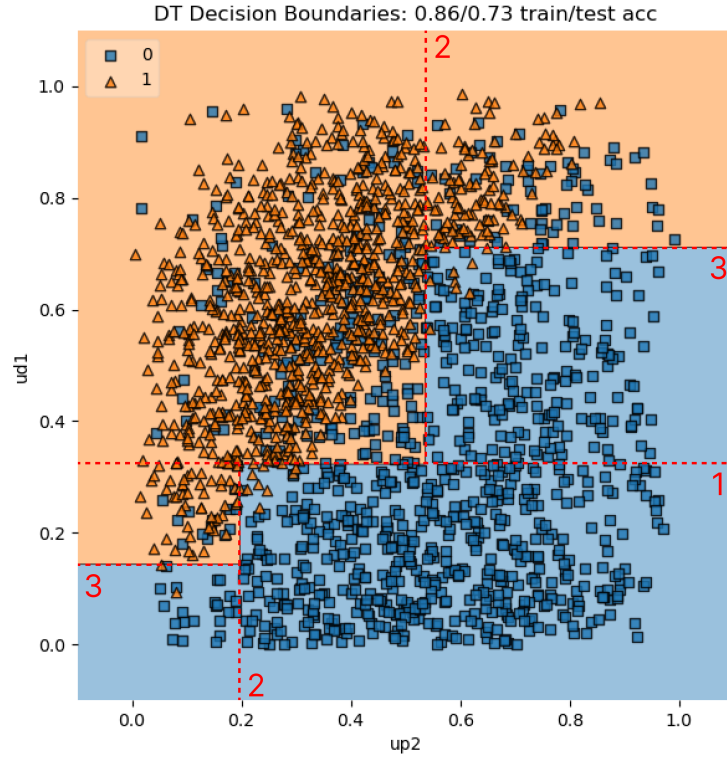


Figure 4.4: Decision Boundaries of a Decision Tree (DT) with depth 3. The numbers in red indicate the DT depth at which a new decision boundary of a subspace in a node is made and the red dotted line shows that decision boundary. The DT is trained to classify classes 'ill defined' (blue square) and 'not ill defined' (orange triangle) using only design parameters u_2^+ and u_1^- .

algorithm that is used often for classification tasks. A fundamental concept of Support Vector Classification (SVC) is the maximization of the margin between clusters of points belonging to different classes. This emphasis on maximizing the margin is aimed at enhancing the SVC's ability to generalize well to unseen data by providing a robust decision boundary. The points closest to the decision hyperplane are called support vectors and they play a big role in deciding the orientation of the hyperplane. However, in our problem this concept is less relevant. This is because when the parameter space is extensively sampled, the margin between the clusters of points belonging to different classes may diminish or even disappear entirely. As a result, the concept of support vectors loses its significance. Instead, our focus shifts towards ensuring a more expressive decision boundary. This thesis will focus on the linear SVC as we are interested in higher order linear relations between the design param-

eters. We implement the linear SVC by using a Python framework using the LIBLINEAR, a linear classification library from the National Taiwan University [15].

Given a set of training samples (X_i, y_i) , where X_i is the input data and y_i is the corresponding class label, the optimal hyperplane decision boundary is defined by

$$\sum_i^N w_i \cdot x_i + b = \vec{w} \cdot \vec{x} + b = 0, \quad (4.8)$$

which is a sum of N (dimension of the hyperplane) products of input data x_i , in our case design parameters and corresponding weights w_i plus a bias term b . The SVC defines the optimal hyperplane as the one that minimises the loss function, and optimises the split parameters $\theta = (\vec{w}, b)$. A common loss function for a linear SVCs is the squared hinge function

$$SH(y, f(x)) = (\max(0, 1 - y * f(x)))^2, \quad (4.9)$$

where y is the true output label and $f(x)$ is the SVCs decision function, both expressed as either +1 (True) or -1 (False).

Where the decision boundary at each node of a DT is just a first order inequality, the decision boundary of a linear SVC is a linear inequality with an order of the number of design parameters, see Equation (4.8).

Figure 4.5 shows the decision boundary generated by a Linear SVC on the same data as in Figure 4.4. It shows that a linear SVC decision boundary not only outperforms a DT of depth 3 in terms of test accuracy, but also reduces over-fitting compared to that DT. Since this example is taken from our real data and we expect most linear relations to not be axis parallel, we expect the SVC decision boundary to be promising when it comes to separating t-graphs in our design parameter space with respect to properties. That is why propose to dress a DT with linear SVC decision boundaries at each node.

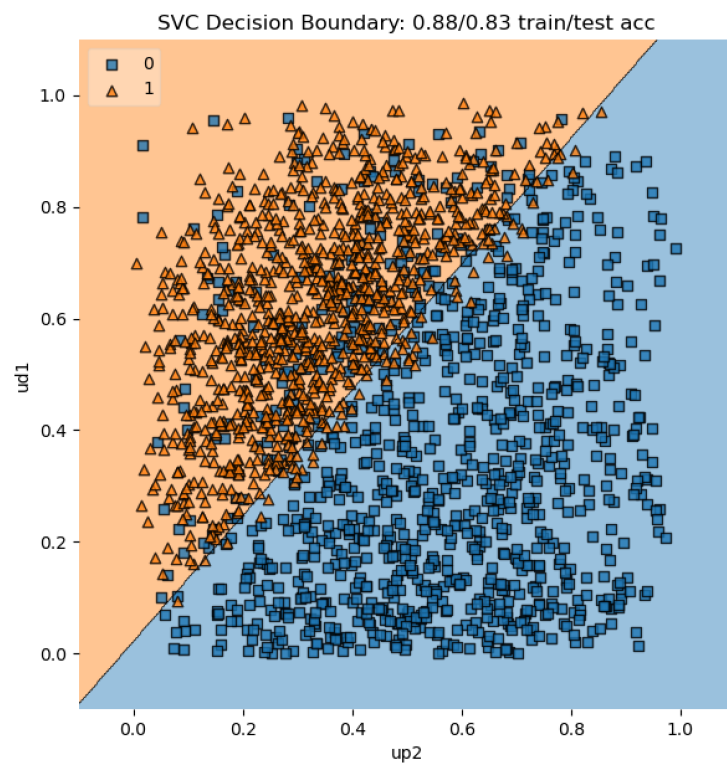


Figure 4.5: Decision boundary of a Linear Support Vector Classifier (SVC). The SVC is capable of generating non-axes parallel decision boundaries.

A Support Vector Classifier inspired Decision Tree

Inspired by oblique (non axis parallel) and non-linear decision trees [16–18] we propose an SVC inspired Decision Tree. Its workflow is shown in Figure 5.1.

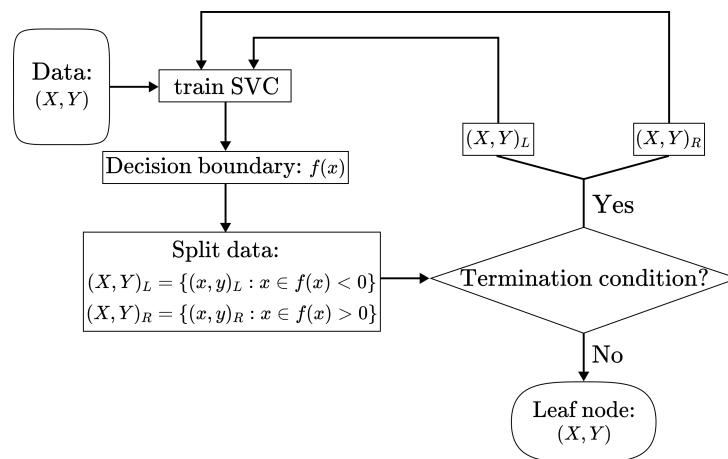


Figure 5.1: Workflow of the SVC inspired DT algorithm. We see how at each node the data is split into two based on the decision boundary. If a termination condition is met, we end up with a leaf node.

Let us introduce the SVC inspired DT algorithm by going through it in steps.

1) Like a decision tree, at each node the SVC inspired DT splits the design

parameter space into two half spaces, but unlike the DT, it does so with a more expressive decision boundary from Equation (4.8). We find this decision boundary by optimising our split parameters $\theta = (\vec{w}, b)$ to minimise (4.9). This gives a decision boundary $f(x)$. We split the data based on this decision boundary:

$$\begin{aligned} (X, Y)_L &= \{(x, y) | x_i \in f(x_i) < 0\} \\ (X, Y)_R &= \{(x, y) | x_i \in f(x_i) > 0\}, \end{aligned} \quad (5.1)$$

where $f(x)$ is the decision boundary as described in Equation (4.8).

2) Then, it checks whether any of the termination conditions are met. We have three of them. They are: I: the depth of the tree exceeds the maximum depth. The maximum depth can be set by the user. II: All instances of data in (X, y) are of 1 class, this means we get a pure node. III: The information gain (4.5) of the classification is 0.

3) If none of the previous conditions are met, we take the split data and initiate a new node on both sides (5.1) of the decision boundary $f(x)$.

This SVC inspired DT, like a regular DT, after training, returns a set of d -dimensional linear inequalities, in the form of the $(d - 1)$ -dimensional decision boundaries with a DT like structure. Recall d , the number of design parameters: $n_h^2 + n_h$. These results resemble those of the DT and are still interpretable.

The data that the SVC inspired DT was trained on was under-sampled to have equal representation of both classes in the classification task as we found the models performance to be best with that distribution of classes.

Results

Here, we present results of the SVC inspired DT, discussed in the previous section, on classification tasks that involve ‘ill definition’ and ‘avalanche(s)’ for a system of $n_h = 2$ interacting hysterons. Then we will present classification results relating to more specific avalanche transitions for $n_h = 2$. Also, we will show the results of classification task for a system of $n_h = 3$ interacting hysterons. finally we discuss the interpretability of the decisions made by the SVC inspired DT.

6.1 2 hysterons

So first let us look how the SVC inspired DT does at the same classification tasks from 4.3(a), namely the classification of ‘ill definition’ and ‘avalanche(s)’.

Figure 6.1 shows improved results compared to a regular DT, see Figure 4.3. We see a decrease in overfitting, implying the model is able to generalise better. Also, the more expressive decision making realises a higher accuracy than the DT with just one decision. Moreover, we find that test accuracy converges at an accuracy of about 0.98 and 0.92 for ‘ill definition’ and ‘avalanche(s)’ respectively, outperforming the DT.

As mentioned before, t-graphs with one or more avalanche(s) come in many forms. For $n_h = 2$, every non-Preisach graph in Fig. 2.2 (c-k) has at least one avalanche. In order to see how the model performs a clas-

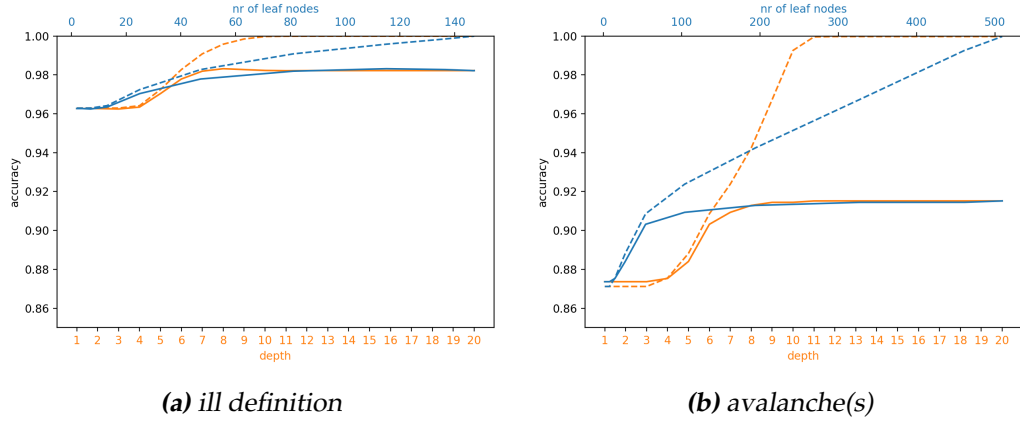


Figure 6.1: Both figures show the train (dashed) and test (solid) accuracy as a function of the depth (orange) and the number of leaf nodes (blue) of the SVC inspired DT for a system of 2 interacting hysterons.

sification task for what we expect to be a smaller subspace in the design parameter space, we looked at t-graphs that include specific avalanche transitions. In particular, the transition $00 \rightarrow 11$ (f-h) and $01 \rightarrow 10$ (c,d).

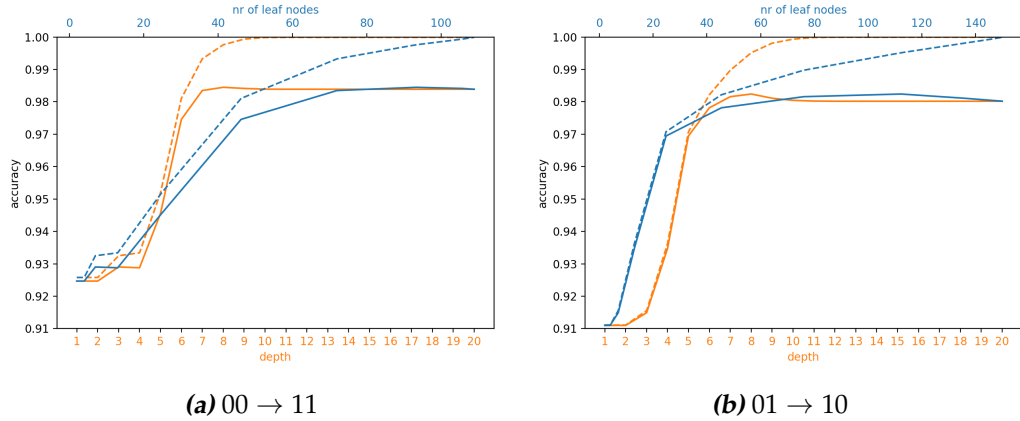


Figure 6.2: Both figures show the train (dashed) and test (solid) accuracy as a function of the depth of the SVC inspired DT for a system of 2 interacting hysterons. On the left the model performance on classification of avalanche transition $00 \rightarrow 11$ and on the right $01 \rightarrow 10$.

Figure 6.2 shows that the SVC inspired DT is able to almost perfectly classify t-graphs containing either avalanche transition. As well as being able to generalise well, as it shows very minimal over-fitting.

Important to note, is that all the datasets for classification results in Figure 6.1 and 6.2 are of the same size, in order to compare the results between classification tasks. For avalanche(s) for example this means that higher accuracies are possible, but require a lot more data. Fig. 6.3 shows that

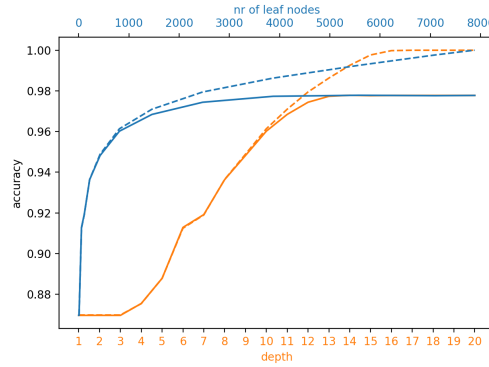


Figure 6.3: Train (dashed) and test (solid) accuracy as a function of the depth (orange) and the number of leaf nodes (blue) of the SVC inspired DT on the classification of avalanche(s) with about 10 times as many data entries as in Fig. 6.1(b).

with about 10 times as many data entries the accuracy converges around an accuracy of 0.97 instead of 0.92 (Fig. 6.1(b)).

We calculated the train- and test accuracies as a function of the depth of the tree. And while the point at which the accuracy converges is a good indicator of how hard the classification is, the number of leaf nodes more directly indicates how hard the sub-space housing t-graphs with common topologies is to classify. Fig. 6.1 shows that the sub-space housing t-graphs with 'avalanche(s)' is harder to classify than the sub-space housing 'ill defined' t-graphs. We state that because the accuracy at convergence is lower for the classification of 'avalanche(s)' than for the classification of 'ill definition' and it does so at a higher number of leaf nodes. We also find that if we look at a property that we expect to correspond to a smaller sub-space housing t-graphs with that property (Fig. 6.2), the same is true.

6.2 3 hystérons

Here, we present the results of the SVC inspired DT on 'ill defined' and 'avalanche(s)' classification tasks.

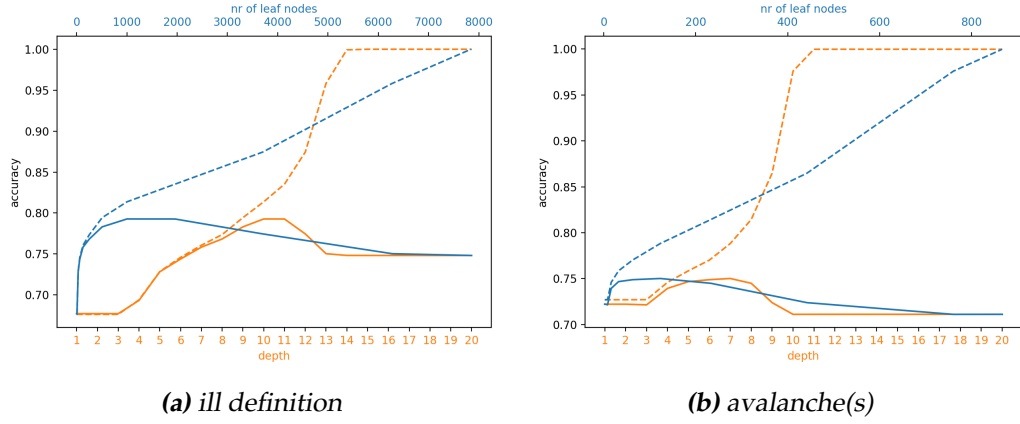


Figure 6.4: Both figures show the train (dashed) and test (solid) accuracy as a function of the depth (orange) and the number of leaf nodes (blue) of the SVC inspired DT for a system of 3 interacting hysterons.

Opposite to the results presented in Sec. 6.1, we see little improvement with respect to both accuracy and mitigating overfitting in Fig. 6.4. We think some of these problems to be the consequence of discrepancies in scale between the design space and the coarse-grained sub-spaces housing common properties. That is to say, we know the design parameter space to increase in complexity with the n_h and expect that the sub-spaces housing t-graphs with common topologies to also become more complex and scattered in the design space. We expect some of these problems to be remedied by even more extensive sampling of the design parameter space.

6.3 Decisions

Recall that the SVC inspired DT also neatly returns a set of order d inequalities (4.8) in a DT like structure that represent $(d - 1)$ -dimensional hyperplanes in the d -dimensional design parameter space. This result facilitates the investigation of the statistical structure of the design parameter space as it provides interpretable rules that delineate the design parameter space according to these t-graph properties. However, although the accuracy appears high after just one decision, as indicated by results presented showing high accuracy at a depth of 1, it becomes evident that accuracy converges at a greater depth, requiring a significantly higher number of leaf nodes. Consequently, interpreting these decisions becomes more challenging. Additionally, as the number of hysterons increase, we anticipate a corresponding growth in the complexity and number of these decisions.

Conclusions and future prospects

7.1 Conclusion

What this thesis hopes to contribute to the exploration of realising inversely designed metamaterials, is an interpretable Machine Learning model that probes the design parameter space of interacting hysterons that model metastable materials made from bistable elements. The model allows us to interpret the decision making, and therefore find relations between the rules of the design parameters and transition graph properties, which helps facilitate understanding of the design parameter space.

The model, a Support Vector Classifier (SVC) inspired Decision Tree (DT), attempts to find patterns in the convoluted design space and does so with a set of interpretable decisions in a tree like structure. We found that the SVC inspired DT is able to almost perfectly capture sub-spaces defined by t-graphs with common properties in the design space. It does so by iteratively splitting the design space into two half-spaces with a decision boundary in the form of a linear inequality (Sec. 5). The aforementioned SVC inspired DT is able to almost perfectly capture common properties between t-graphs in the design space for $n_h = 2$ with a collection of decision boundaries in the form of linear inequalities between design parameters (Sec. 6). Thus implying that higher order linear relations than pairwise linear relations [5, 8] exist, that are able to describe common properties between t-graphs. These linear relations are interpretable which helps facilitate understanding of the design parameter space.

7.2 Discussion

Even though for $n_h = 2$ the model is able to almost perfectly capture sub-spaces housing t-graphs with common properties, the same can not be said for $n_h = 3$. Even though it is able to learn the subspace in training, it overfits, resulting in a result that is not general.

We do expect some of these problems to be remedied by even more extensive sampling of the design parameter space, so as to reconcile the predicted discrepancies in scale between the design space and the coarse grained sub-spaces housing t-graphs with common properties.

Additionally, we also hypothesise that the model has difficulty isolating sub-spaces housing t-graphs with common properties that are a collection of scattered clusters in the design parameter space. We suspect that performing clustering can help the model in these cases, since it would allow each of these clusters to be considered separately by the model instead of all at once.

It is also interesting to note that in the exploration for $n_h > 2$ the linear SVC scales almost linearly with the number of data instances, as well as the dimension of each instance d (the number of design parameters). This means that the linear SVC scales quadratically with the number of hysterons n_h . However, since we expect these classification tasks to be harder for higher n_h , and thus require more decision nodes, but do not know explicitly how the number of nodes is related to the number of interacting hysterons in the system n_h , we can only say that the model will scale at least quadratically with the n_h .

7.3 Future Prospects

One way to be more informed about the design space is to investigate the statistical structure of the design parameter space and in particular of the sub-spaces that are defined by transition graph properties. The model presented in this thesis facilitates this investigation as it provides interpretable rules that delineate the design parameter space according to these t-graph properties.

Between the 1- and d -dimensional linear inequalities of the DT and SVC inspired DT respectively, it would be interesting to see if any d' -dimensional

linear relations exist, where $1 < d' < d$. Realising this, by changing the decision boundary at each node to compare all combinations of d' number of design parameters and then choose the best split, has proven technically difficult. It would be especially interesting to see if we can use it to find a complete set of pairwise inequalities for $n_h = 2$ and $n_h = 3$ as a performance benchmark.

Bibliography

- [1] J. Robertson. Amorphous carbon. *Advances in Physics*, 35(4):317–374, Oct 1986.
- [2] Zbigniew H. Stachurski. On structure and properties of amorphous materials. *Materials*, 4(9):1564–1598, Sep 2011.
- [3] Muamer Kadic, Graeme W. Milton, Martin van Hecke, and Martin Wegener. 3d metamaterials. *Nature Reviews Physics*, 1(3):198–210, Jan 2019.
- [4] F. Preisach. Über die magnetische nachwirkung - zeitschrift für physik a hadrons and nuclei, Feb 1935.
- [5] Martin van Hecke. Profusion of transition pathways for interacting hysterons. *Physical Review E*, 104(5), nov 2021.
- [6] M. Mert Terzi and Muhittin Mungan. State transition graph of the preisach model and the role of return-point memory. *Phys. Rev. E*, 102:012122, Jul 2020.
- [7] Hadrien Bense and Martin van Hecke. Complex pathways and memory in compressed corrugated sheets — pnas, Dec 2021.
- [8] Margot Teunisse and Martin van Hecke. Constructing the transition graphs of interacting hysterons. *unpublished*, 2024.
- [9] Liwei Wang, Yu-Chin Chan, Faez Ahmed, Zhao Liu, Ping Zhu, and Wei Chen. Deep generative modeling for mechanistic-based learning and design of metamaterial systems. *Computer Methods in Applied Mechanics and Engineering*, 372:113377, December 2020.

- [10] M. Bessa, Piotr Glowacki, and Michael Houlder. Bayesian machine learning in metamaterial design: Fragile becomes supercompressible. *Advanced Materials*, 31:1904845, 10 2019.
- [11] Gabriele M. Coli, Emanuele Boattini, Laura Filion, and Marjolein Dijkstra. Inverse design of soft materials via a deep learning–based evolutionary strategy. *Science Advances*, 8(3):eabj6731, 2022.
- [12] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and regression trees*. The Wadsworth Statistics/Probability Series. Belmont, California: Wadsworth International Group, a Division of Wadsworth, Inc. X, 358 p., 1984.
- [13] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
- [14] Sreerama K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2(4):345–389, 1998.
- [15] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [16] K.P. Bennett and J.A. Blue. A support vector machine approach to decision trees. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, volume 3, pages 2396–2401 vol.3, 1998.
- [17] Vlado Menkovski, Ioannis T. Christou, and Sofoklis Efremidis. Oblique decision trees using embedded support vector machines in classifier ensembles. In *2008 7th IEEE International Conference on Cybernetic Intelligent Systems*, pages 1–6, 2008.
- [18] Rodrigo Barros, Ricardo Cerri, Pablo Andretta Jaskowiak, Andre de Carvalho, and A. F. A bottom-up oblique decision tree induction algorithm. 11 2011.