Using Kan Extensions to Motivate the Design of a Surprisingly Effective Unsupervised Linear SVM on the Occupancy Dataset

Matthew Pugh Jo Grundy Nick Harris
School of Electronics and Computer Science
University of Southampton
mp8g16@soton.ac.uk j.grundy@soton.ac.uk nrh@ecs.soton.ac.uk

Recent research has suggested that Category Theory can provide useful insight into the field of Machine Learning (ML). One example is improving the connection between an ML problem and the design of a corresponding ML algorithm. A tool from Category Theory called a Kan extension is used to derive the design of an unsupervised anomaly detection algorithm for a commonly used benchmark, the Occupancy dataset. Achieving an accuracy of 93.5% and a ROCAUC of 0.98, the performance of this algorithm is compared to state-of-the-art anomaly detection algorithms tested on the Occupancy dataset [2]. These initial results demonstrate that Category Theory can offer new perspectives with which to attack problems, particularly in making more direct connections between the solutions and the problem's structure.

1 Introduction

Category theory is not a discipline traditionally known for its practical applications. However, there have been indications that it can benefit the field of machine learning [15]. In particular, Kan extensions, a tool from Category theory, have been used to describe the construction of a handful of supervised learning algorithms [14]. This paper will look at applying this construction to motivate the design of an unsupervised classification algorithm. Seeking to more closely link the outcomes of the data analysis to the design of the algorithm on a real-world problem.

The Occupancy dataset, first introduced for supervised learning, has also been used to demonstrate the performance of unsupervised anomaly detection algorithms on time series data. Though a seemingly reasonable choice for this task, it possesses two potential issues. Firstly, its classification labels have a limited relationship to the information provided by the timestamps or sequence of recorded points. Secondly, the dataset contains anomalous points whose nature is not reflected in their classifications. These characteristics are indicated by the initial data analysis and included in the construction of a Kan Extension which is used to derive the "Constrained SVM" (C-SVM) algorithm.

The C-SVM is an unsupervised linear SVM whose hyperplane is constrained to intersect a given point. It achieved an accuracy of 93.5% and a ROCAUC of 0.98. A competitive performance when compared to algorithms presented in related works. Motivating the design of C-SVM from the hypothesised characteristics of the Occupancy dataset allows its performance to validate their relevance to the machine learning problem. The presence of these characteristics indicates that the dataset should be used with caution when bench-marking other time-series anomaly detection algorithms.

Submitted to:
ACT 2023

2 Background

The Occupancy dataset is a five-dimensional time series dataset that records temperature (Celsius), relative humidity (Percentage), light (Lux), CO2 (parts per million), Humidity Ratio (kilograms of water divided by kilograms of air) in an office room. Each data point has one of two classes which indicates if the room is occupied or not [2].

Introduced by Luis M. Candanedo and Véronique Feldheim (accessible at UCI ML repository as of 2022/01/16), the original paper explores the use of supervised machine learning models to detect building occupancy and improve the energy efficiency of smart buildings.

The dataset contains 20,560 data points recorded over 16 days. 15,843 (77.1%) of the points correspond to not-occupied and 4717 (22.9%) to occupied. The bias towards the not-occupied class has led to the dataset being used to evaluate unsupervised classification techniques.

This dataset has been used in a large number of works. For comparison, these are limited to: English language primary research, which tests an unsupervised algorithm to classify occupancy on the unmodified occupancy dataset.

Screening occurred in two phases, the first based only on the abstract and the second considering the full paper. From an initial pool of 226 papers that cited this dataset, 12 met the criteria [1, 6, 10, 17, 16, 7, 18, 9, 4, 3, 8, 12].

PCA was used to plot the first three principal components of the dataset. The timestamps for each point were removed, and their sequential nature was ignored (Fig 1).

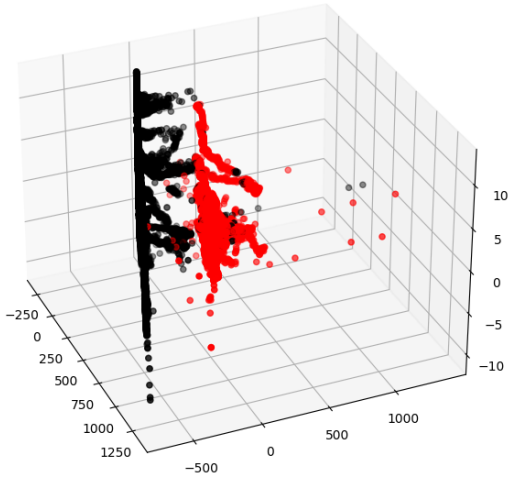


Figure 1: The first three principle components of the Occupancy dataset produced by the PCA algorithm after the data was normalised to have a mean of zero and variance of one in each dimension. The time component of the data was ignored. Points corresponding to the room being unoccupied are coloured black, and occupied are coloured red.

The figure shows a clear separation between the two classes of the occupancy dataset, indicating that a hyperplane may suitably classify it. As this separation is seen when time is disregarded, it would appear that the temporal component of the dataset is not a particularly relevant feature for the classification problem, and should be disregarded by a potential classification algorithm. It is also helpful to note that a small number of points from both classes deviate from the main body of the data. This appears to be due to one of the sensors breaking during data collection.

3 Kan Extensions for Classification Algorithms

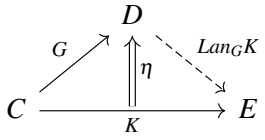
Previous works have suggested that Kan extensions, a tool from Category theory, might be used to generalise the notion of extrapolating new data points from previous observations, providing an interesting construction for a supervised classification algorithm from Kan extensions [14]. This construction can be modified to help motivate the unsupervised C-SVM algorithm in the following section. An introduction to Category Theory is beyond the scope of this paper, but the following resources can provide it [11, 5].

3.1 Kan Extensions

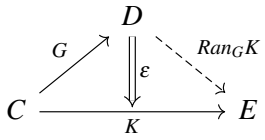
Kan extensions ask how one might extend one functor to produce another. Given two functors $K : C \to E$ and $G : C \to D$, a Kan extension attempts to find a functor $F : D \to E$ such that FG is approximately equal to K. It is overly restrictive (and often less helpful) to ask for FG to be exactly equal to K. So Kan extensions weaken equality to the requirement for some universal natural transformation between the two functors. The left Kan extension asks for a natural transform $\eta : K \Rightarrow FG$, and the right asks for $\varepsilon : FG \Rightarrow K$. The Kan extensions require that for any natural transform γ and functor H pair, the natural transform can be factored uniquely as a composition of the "best" natural transform, and some other natural transform e.g. $\gamma = \alpha\eta$. The notation for the functors which satisfy the requirements of the left and right Kan extensions are $Lan_G K$ and $Ran_G K$, respectively.

The following defines the left and right Kan extension [13].

Definition 3.1 (Left Kan Extension)**.** Given functors $K : C \to E$, $G : C \to D$, a left Kan extension of K along G is a functor $Lan_G K : D \to E$ together with a natural transformation $\eta : K \Rightarrow (Lan_G K)G$ such that for any other such pair $(H : D \to E, \gamma : K \Rightarrow HG)$, γ factors uniquely through η.



Definition 3.2 (Right Kan Extension)**.** Given functors $K : C \to E$, $G : C \to D$, a right Kan extension of K along G is a functor $Ran_G K : D \to E$ together with a natural transformation $\varepsilon : (Ran_G K)G \Rightarrow K$ such that for any $(H : D \to E, \delta : HG \Rightarrow K)$, δ factors uniquely through ε



3.2 A Supervised Classification Algorithm from Kan Extensions

A dataset can be represented by a discrete category I' alongside functors which assign values to each data point. The input data can be described by a functor $G : I' \to I$ (Fig 2). In order to encode some of the geometric information present within the dataset, rather than a discrete category, I is allowed to be a Preorder. An example would be the ordered real numbers $\mathbb{R}_{\leq}$, whose objects are the real numbers and for which a unique morphism $\leq : x \to y$ exists if and only if $x \leq y$.

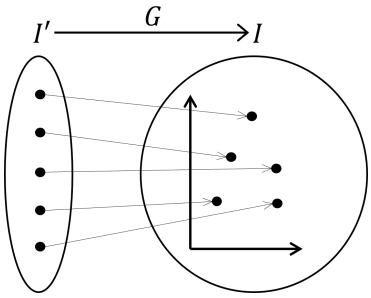


Figure 2: A functor $G : I' \to I$ embedding discrete data points from I' into a richer space I.

For a dataset with binary classification labels, the target data can be represented by a functor $K : I' \to \{false, true\}$. In this instance, $\{false, true\}$ represents an ordered two-object category whose only non-identity morphism is $\leq$: $false \to true$.

For each of the points in I selected by G there is information about their classification labels given by K. The general principle of a supervised classification algorithm is to extend information from a subset to the whole space. This can be described in this context as finding a suitable functor $F : I \to \{false, true\}$.

An initial attempt may be to assign F to be either the left $(Lan_G K)$ or right $(Ran_G K)$ Kan extensions in equations 1,2,3 from [14].

$$Lan_G K : I \to false, true \quad Ran_G K : I \to false, true \tag{1}$$

$$Lan_G K(x) = \begin{cases} true & \exists x' \in I', G(x') \leq x, K(x') = true \\ false & else \end{cases} \tag{2}$$

$$Ran_G K(x) = \begin{cases} false & \exists x' \in I', x \leq G(x'), K(x') = false \\ true & else \end{cases} \tag{3}$$

An example visualisation of these equations can be seen in Fig 3, in which I has been set to $\mathbb{R}_{\leq}$. The figure presents the resulting Kan extensions from datasets with overlapping and non-overlapping classes.

In the case where I is $\mathbb{R}_{\leq}$, F is forced to become a step function due to the induced ordering of the two categories by their morphisms. This creates a decision boundary at some point in $\mathbb{R}_{\leq}$.

$$F : \mathbb{R}_{\leq} \to \{false, true\}$$
$$F(x) := \begin{cases} true & \alpha \leq x \\ false & else \end{cases}$$

For two functors $F, F' : \mathbb{R}_{\leq} \to \{false, true\}$ a natural transform $\gamma : F \Rightarrow F'$ must select for each object in $\mathbb{R}_{\leq}$ a morphism in $\{false, true\}$. This, at most, may alter the output of F from $false$ to $true$ while retaining its monotonicity. Considering the decision boundary α in F, the effect of γ can only be to increase α. This means that a natural transform can only exist between F and F' if $\alpha \leq \alpha'$. When composed with $G : I' \to \mathbb{R}_{\leq}$, the objects of I and their image under G restrict the components of γ. Consequently, the left and right Kan extensions produce classifying functions with no false negatives and no false positives, respectively.

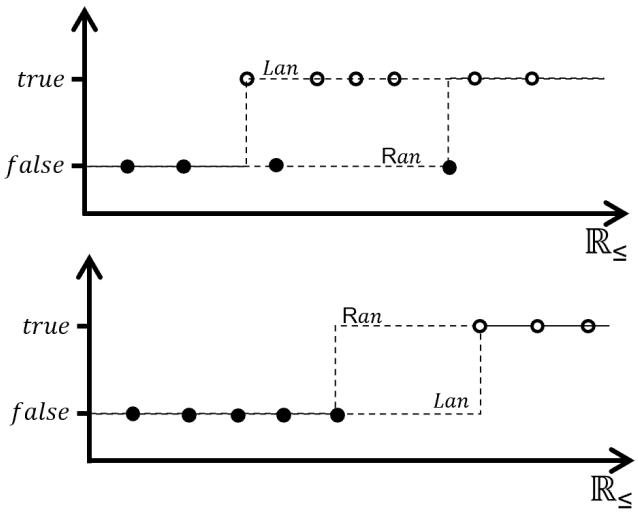


Figure 3: The left and right Kan extensions, $Lan_G K$ and $Ran_G K$ produced from the functors $K : I' \to \{false, true\}$ and $G : I' \to \mathbb{R}_{\leq}$, which represent a binary classification dataset over the ordered real numbers. The two graphs show the different extensions produced from a dataset with overlapping classes (upper) and separable classes (lower)

This approach isn't yet sufficient for more complex systems. To extend the utility of this representation, an additional, trainable, functor $f : I \to I^*$ can be added.

$$\begin{array}{ccc} I & \xrightarrow{f} & I^* \\ {\scriptstyle G}\uparrow & & \downarrow{\scriptstyle F} \\ I' & \xrightarrow{K} & \{false, true\} \end{array}$$

For the case of overlapping classification regions, it is a reasonable assumption that the less these regions overlap, i.e. the smaller the disagreement region, the better the resulting classification is likely to be. For this purpose, by assuming that $I*$ is $\mathbb{R}^d$, a function known as the ordering loss can be introduced [14], with the guarantee that minimizing the ordering loss will also minimize the disagreement region.

$$l : (I \to \mathbb{R}^d) \to \mathbb{R} \tag{4}$$

$$l(f) = \sum_{i \leq a} max(0, max\{f(x)[i] | x \in I', K(x) = false\} \tag{5}$$
$$- min\{f(x)[i] | x \in I', K(x) = true\})$$

Where $f(x)[i]$ is the i-th component of the vector $f(x) \in \mathbb{R}^d$

4 Motivating C-SVM through Kan Extensions

The dataset characteristics presented in section 2, for a supervised problem, would suggest a two-class linear SVM. For an unsupervised problem, a linear SVM will attempt to maximise its distance from all data points, tending towards a hyperplane at infinity. The implemented model is constrained to hyperplanes that pass through a given point. This model will be referred to as the Centred SVM (C-SVM).

Two steps are required to motivate C-SVM from the construction shown in section 3.2. Firstly, details about the dataset which have been introduced in section 2 can be used to populate the construction with information specific to this task. Secondly, the construction of a classification algorithm through Kan extensions needs to be modified for it to define an unsupervised algorithm.

The morphism $G : I' \to I$ assigns to each of the data points in the discrete category I' information regarding the input data of the dataset. In this case, the measured values in the time series can be represented as a five dimensions real number vector $\mathbb{R}^5$. The time series information was determined to be irrelevant in the data analysis. Selecting I' as the discrete category $[n]$ with n objects, acting analogously as unique ids to the n data points in the dataset, $[n]$ disregards any time series information. For the sake of this formulation, rather than using G directly, the data can be shifted to have a mean of $\vec{p}$ giving $G' : [n] \to \mathbb{R}^5$. The choice of $\vec{p}$ is arbitrary, making it a hyper-parameter of this algorithm. By inspection, the data analysis indicates that normalising the datasets to have a mean of zero is a reasonable choice.

The data analysis identified that the data could be suitably separated with a hyperplane. This can be represented by constraining the trainable portion of the construction to be a linear map into the real numbers $f : \mathbb{R}^5 \to \mathbb{R}_{\leq}$. For this construction, $\mathbb{R}^5$ is the discrete category, with $\mathbb{R}_{\leq}$ being the category which represents the ordered set of real numbers. The choice of $\mathbb{R}_{\leq}$ as the codomain of f is equivalent to stating the belief that the points of $\mathbb{R}^5$ can be ordered based on how likely they are to be classified as either "Occupied" or "Non-Occupied". This means that f uses the ordering of $\mathbb{R}_{\leq}$ to induce a partial ordering on the points of $\mathbb{R}^5$ based on their presumed classification. The role of the Kan extensions in this construction is to decide the cutoff, where points greater than a certain value must be classified as "occupied" and less than that value as "not-occupied". These choices result in the following diagram..

$$\begin{array}{ccc} \mathbb{R}^5 & \xrightarrow{\quad f \quad} & \mathbb{R}_{\leq} \\ {\scriptstyle G'}\uparrow & & \vdots\,{\scriptstyle F} \\ [n] & \xrightarrow{\quad K \quad} & \{false, true\} \end{array}$$

The second problem is modifying the construction to be applied to an unsupervised learning problem. The definition of the Kan extension requires that the morphism $K : [n] \to \{false, true\}$ is known. By introducing an additional morphism (eq 6) it is possible to define K through composition $K := (0 \leq) f G'$

$$(0 \leq) : \mathbb{R}_{\leq} \to \{false, true\} \tag{6}$$

$$(0 \leq)(x) := \begin{cases} true & 0 \leq x \\ false & else \end{cases} \tag{7}$$

Interpreting this composition, introducing $(0 \leq)$ converts the hyperplane f into a binary classifier. The points' classification depends on which side of the hyperplane they lie on. Due to the definition of K by composition, it is guaranteed that there is no overlap in the classification boundaries. As the resulting Kan extensions are from the function space $\mathbb{R}_{\leq} \to \{false, true\}$ the resulting left and right Kan extensions

correspond with the functions shown in Fig 3. The decision boundaries formed by the three function $(0 \leq)f$, $(Lan_{fG'}K)f$ and $(Ran_{fG'}K)f$ can be visualised as in Fig 4.

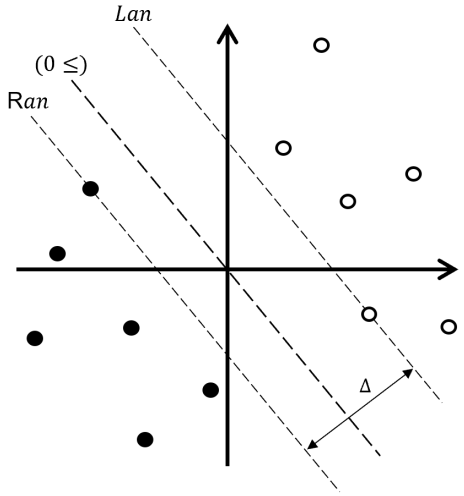


Figure 4: A representation of the classification boundaries produced by the functions $(0 \leq)f$, $(Lan_{fG'}K)f$ and $(Ran_{fG'}K)f$ for an imagined dataset on the plane.

Any choice of f now produces a preliminary classification of the points in the Occupancy dataset. The left and right Kan extensions identify the points closest to the decision boundary. The classification quality produced by a given f can be judged by the distance between the left and right Kan extensions, Δ.

The ordering loss, as previously defined, cannot be used directly for this version of the problem, as it is zero when there is no overlap between classification regions. However, given that it can be guaranteed that there will never be an overlap of the classification regions, the outer max function of the ordering loss function can be removed, allowing the modified ordering loss function (l') to become negative (eq 8,9). Minimisation of the modified ordering loss function maximises the separation region between the left and right Kan extensions. In the particular case where f has codomain $\mathbb{R}_{\leq}$, the modified ordering loss reduces to be $l'(f) = -\Delta$, where Δ can be seen as the difference between the closest points on each side of the hyperplane f when projected down onto the real numbers (Fig 4). Ultimately, the choices and modifications applied to the construction produce an algorithm which appears to be a linear, unsupervised SVM whose hyperplane is constrained to pass through $\vec{p}$

$$l' : (I \to \mathbb{R}^d) \to \mathbb{R} \tag{8}$$

$$l'(f) = \sum_{i \leq a} max\{f(x)[i] | x \in I', K(x) = false\} \tag{9}$$
$$- min\{f(x)[i] | x \in I', K(x) = true\}$$

5 Implementation of C-SVM

From the construction presented in the previous section, the remaining task is to produce an algorithm which finds a suitable, linear transform, $f : \mathbb{R}^5 \to \mathbb{R}_{\leq}$, which maximises the distance (Δ) between the decision boundaries produced by the left and right Kan extensions. As f is a linear transform, it can be defined as $f(\vec{x}) := \vec{x} \cdot \hat{v}$, where $\hat{v}$ is a five-dimensional real number vector. Applying f to every point of the normalised dataset, Δ can be computed as the difference between the data point with the smallest positive value after f, and the data point with the largest negative value after f. For the purposes of classification, the process of normalising the dataset and then transforming with f can be understood as assigning a score to each point based on its signed distance from the hyperplane. This corresponds with the image of the point in $\mathbb{R}_{\leq}$ (Equation 10):

$$\text{score} = (\vec{x} - \vec{p}) \cdot \hat{v} \tag{10}$$

The unsupervised fitting algorithm (Algorithm: 1), which this paper introduces, determines the value for each dimension of the normal vector in turn, by rotating the vector around axes of a hyper-sphere centred at the constraining point. For each iteration, there are set values, the current value, and the unset values. For each loop, the fitting algorithm checks an integer number of uniformly spaced angles (set by the parameter "res") from the interval $[0, \pi)$ to determine the current value, which maximises the plane's distance to the closest point. The set values remain unchanged, and only the current value and unset values are varied. At the end of the loop, the maximising value is assigned to the current value.

$$\text{Angle2Value}(\hat{v}, i, j, \theta) = \begin{cases} \hat{v}_j & j < i \\ \sin(\theta) & j = i \\ \frac{\cos(\theta)\sqrt{1-\sin(\theta)}}{dim(\hat{v})-i} & j > i \end{cases} \tag{11}$$

Algorithm 1 Fitting C-SVM

Input: data $\in \mathbb{R}^{n \times m}$, res $\in \mathbb{N}$
Output: $\vec{\mu}, \hat{v} \in \mathbb{R}^m$

$\vec{\mu} \leftarrow$ mean of the n points in data
$\hat{v}_{a \leq m} \leftarrow 0$
for $i \leftarrow 0$ **to** $i = m$ **do**
    $\Theta_{a<res} \leftarrow a\pi/(\text{res}-1)$
    $\text{V}_{\text{j}\leq\text{m, a}<\text{res}} \leftarrow \text{Angle2Value}(\hat{v}, i, j, \Theta_a)$
    $\text{S} \leftarrow \text{MatrixProduct}(data - \vec{\mu},\ \text{V})$
    $\text{L} \leftarrow \text{MinAlongFirstAxisIfTrue}(\text{S}, \text{IsPositive})$
    $\text{R} \leftarrow \text{MaxAlongFirstAxisIfTrue}(\text{S}, \text{IsNegative})$
    $\Delta \leftarrow \text{L} - \text{R}$
    $k \leftarrow \text{MaxIndex}(\Delta)$
    $\hat{v}_i = V_{i,k}$
end for
return $\vec{\mu}, \hat{v}$

By fitting each of the values sequentially, the fitting algorithm achieves a time complexity of $O(nm^2\,\text{res})$ with n being the number of data points, m being the number of dimensions and res being the resolution of the angle search space.

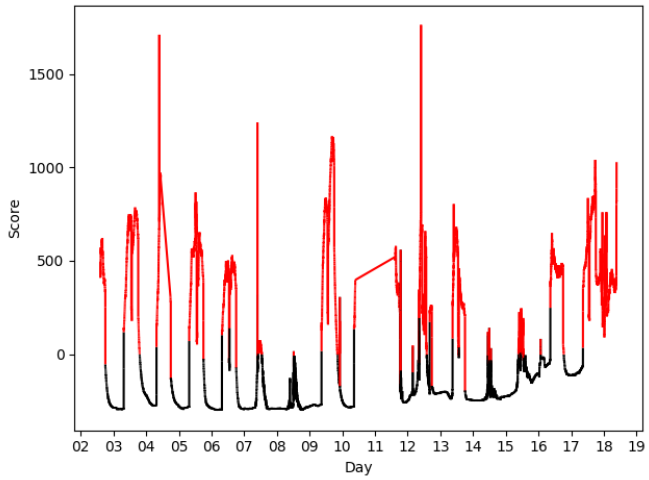


Figure 5: The scores generated by the fitted C-SVM on the Occupancy dataset. Points which are labelled as occupied in the dataset are coloured red, and unoccupied are coloured black.

6 Results

The scores given by the fitted C-SVM on every point in the dataset are shown in Fig 5.

Points that the Occupancy dataset labels as occupied have been coloured red, and the points corresponding to not-occupied have been coloured black. By visual inspection, higher scores assigned by C-SVM correlate strongly with the building being occupied. The Receiver Operator Curve (ROC) for these scores, when compared to the ground truth labels of the dataset, are shown in Fig 6. The model achieved a ROCAUC of 0.9814 with a maximum accuracy of 93.5%.

Four of the papers found in Section 2 provided ROCAUC scores for models tested on the Occupancy dataset. The C-SVM algorithm achieved the fourth highest ROCAUC out of twenty-six models, with a difference in ROCAUC of 0.0146 between it and the top-performing model (Appendix A). Six of the papers provided classification accuracies (with varying levels of precision) for tested models. The C-SVM algorithm had the sixteenth highest accuracy out of twenty-eight models, with a difference in the accuracy of 3.5% between it and the top-performing model (Appendix A)

7 Discussion

7.1 Category Theory

Commonly, intuition and experience are used to bridge the gap between knowledge about an ML problem and the design of an algorithm. Category Theoretic techniques have the potential to make this bridge more explicit. Using the template of the Kan Extension, adding information about the dataset began to outline the necessary algorithm. Not only does this form of notation highlight the reasoning for specific choices but it also begins to suggest an alteration to the classic ML design loop. Traditionally, by testing iterations of algorithms, an engineer may understand more about the particular problem they are working on. Intuitively this means adding information about the dataset into the design of the algorithm. However, with the categorical perspective, it may be more sensible to add new information about the structure of a

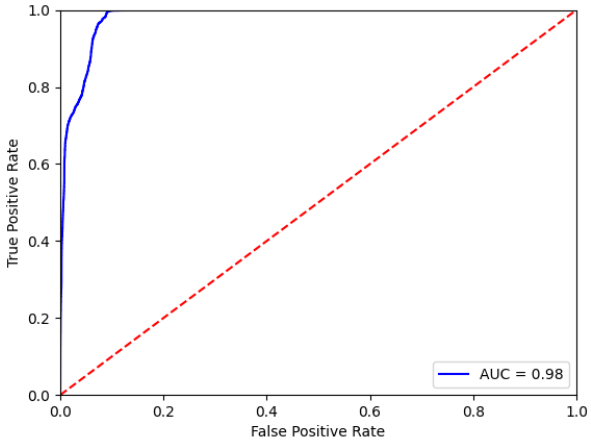


Figure 6: The Receiver Operator Curve (ROC) for the scores given by the C-SVM when compared to the ground truth labels in the occupancy dataset. The model achieved a ROCAUC of 0.9814.

dataset to its categorical description, trusting that these changes will indicate the appropriate modification to the algorithm design.

It is worth noting that the techniques used to derive the C-SVM algorithm are relatively crude compared to what may be possible with Category Theory. It was necessary to work inside of the category of Categories to make use of the natural transforms required in the definition of the Kan extension. Unfortunately, this led to the use of categories themselves as the objects of interest. It is true that categories can encode rich mathematical structures, but it is often their objects which represent the structures. For example, vector spaces would have been useful in the definition of C-SVM and do exist as objects inside of categories, but are not necessarily categories themselves. Future development of the techniques shown may benefit from more nuanced constructions to increase their flexibility and descriptive power.

7.2 The Occupancy Dataset and Anomaly Prediction

The performance of the C-SVM algorithm highlights two issues with using the Occupancy dataset as a benchmark for anomaly prediction algorithms.

The first concern comes from its use in evaluating the performance of algorithms on time series data. 10 of the 12 included papers tested time-sensitive algorithms. However, the PCA analysis in Fig 1 and the ROCAUC of the C-SVM algorithm raise the concern that effectively, none of the features required for successful classification are present in the temporal information. C-SVM was outperformed in accuracy by fifteen of the algorithms presented in relevant papers. However, with an accuracy difference of only 3.5%, it becomes unclear whether this improvement is due to the time-series information. For the purpose of investigating the performance of an algorithm on a time series dataset, this uncertainty impairs the utility of the Occupancy dataset as a benchmark. If the accuracy difference is entirely due to time-series information, such a slight variation reduces the resolution of the dataset in differentiating the attributes of tested algorithms. This property of the dataset may lead to improper evaluation of model performance when not accounted for: either by over-representing the performance of an algorithm which ineffectively utilises time series information or in the underperformance of models such as LSTMs, whose additional connection weights increase the dimensionality and symmetries of their loss plane without conferring

any significant benefit in this case.

The second concern can be seen in the data points generated by a broken sensor. These data points occur for both occupied and not-occupied classifications, but considering the use of anomaly detection algorithms, it creates an issue. The Occupancy dataset in this context is a dataset with three classifications: occupied, not-occupied, and true anomaly. For this reason, the labels included in the dataset cannot be taken as ground truth classifications when validating anomaly detection algorithms. An algorithm may correctly identify the erroneous points as anomalies but be punished in its resultant score. In this way, it should be considered that the performance of such algorithms on this dataset are not an entirely accurate indication of their performance as anomaly detection systems. The C-SVM algorithm, due to its simplicity, is largely insensitive to the more nuanced forms of anomalies created by the broken sensor, contributing to its outperformance of comparatively more advanced systems.

The concerns identified with the Occupancy Dataset do not necessarily mean it should be completely disregarded. Only in the cases of validating algorithms which are sensitive to time series information, or which identify anomalies, should these concerns be considered in their performance. In summary, it is suggested that the Occupancy Dataset should be used with caution.

7.3 Uses of the Centred SVM

Though this paper has primarily utilised the C-SVM to demonstrate certain properties of the Occupancy dataset, it may also provide value in other applications. Its low time complexity allows it to be implemented as a component of a larger system, or as another tool for data analysis. The ability to control the constraining point provides the opportunity for manual selection, or for the point to be provided by another algorithm. Furthermore, the optimising algorithm presented may use any loss function relative to the dataset. Situations, where the separation between datasets is less clear, may cause issues as the hyperplane which maximises the distance to the nearest point may not generate a suitable result. Alterations such as using the distance to the k-th nearest point, the average distance of k points, or maximising the distance variance (in which its operation becomes similar to the Fischer linear discriminant) may all be varied on a case-by-case basis.

8 Conclusion

It has been demonstrated that it is possible to describe a simple unsupervised anomaly detection algorithm, for use on a real world problem, with Kan extensions. As a result, the design of the algorithm was directly informed by characteristics of the dataset, discovered in the initial data analysis. Not only was this algorithm competitive with those presented in related works, its Kan extension inspired design was able to add supporting evidence to claims about characteristics of the Occupancy dataset. Namely, that its classification labels have little relationship to the temporal component of the data and that truly anomalous data points exist whose nature is not reflected by their classification. Characteristics which should be considered before utilising the Occupancy dataset as a benchmark for other algorithms.

The development of Category theoretic techniques may ultimately generate useful tools for the construction of machine learning algorithms. Providing a perspective which is more concerned with the structure of data than the particular implementations of algorithms. Though Kan extensions have provided a promising indication of what these techniques may look like, there are many facets which future work may improve.

Acknowledgements and Disclosure of Funding

This work was partly funded by the grant "Early detection of contact distress for enhanced performance monitoring and predictive inspection of machines" (EP/S005463/1) from the Engineering and Physical Sciences Research Council (EP- SRC), UK, and Senseye. C. Cirstea is thanked for her helpful discussions.

References

[1] Luis M. Candanedo, Veronique Feldheim & Dominique Deramaix (2017): *A methodology based on Hidden Markov Models for occupancy detection and a case study in a low energy residential building*. *ENERGY AND BUILDINGS* 148, pp. 327–341, doi:10.1016/j.enbuild.2017.05.031.

[2] Luis M. Candanedo & Véronique Feldheim (2016): *Accurate occupancy detection of an office room from light, temperature, humidity and CO 2 measurements using statistical learning models*. *Energy and Buildings* 112, pp. 28–39, doi:10.1016/j.enbuild.2015.11.071. Available at `https://linkinghub.elsevier.com/retrieve/pii/S0378778815304357`.

[3] Tolga Ergen & Suleyman S. Kozat (2020): *A novel distributed anomaly detection algorithm based on support vector machines*. *DIGITAL SIGNAL PROCESSING* 99, doi:10.1016/j.dsp.2020.102657.

[4] Tolga Ergen & Suleyman Serdar Kozat (2020): *Unsupervised Anomaly Detection With LSTM Neural Networks*. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS* 31(8), pp. 3127–3141, doi:10.1109/TNNLS.2019.2935975.

[5] Brendan Fong & David I. Spivak (2018): *Seven Sketches in Compositionality: An Invitation to Applied Category Theory*, doi:10.48550/arXiv.1803.05316. Available at `http://arxiv.org/abs/1803.05316`. Number: arXiv:1803.05316 arXiv:1803.05316 [math].

[6] Xiaowei Gu, Plamen Angelov, Dmitry Kangin & Jose Principe (2018): *Self-Organised direction aware data partitioning algorithm*. *INFORMATION SCIENCES* 423, pp. 80–95, doi:10.1016/j.ins.2017.09.025.

[7] Xiaowei Gu, Plamen Angelov & Zhijin Zhao (2019): *A distance-type-insensitive clustering approach*. *APPLIED SOFT COMPUTING* 77, pp. 622–634, doi:10.1016/j.asoc.2019.01.028.

[8] Mine Kerpicci, Huseyin Ozkan & Suleyman Serdar Kozat (2021): *Online Anomaly Detection With Bandwidth Optimized Hierarchical Kernel Density Estimators*. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS* 32(9), pp. 4253–4266, doi:10.1109/TNNLS.2020.3017675.

[9] Anna Khalemsky & Roy Gelbard (2020): *A dynamic classification unit for online segmentation of big data via small data buffers*. *DECISION SUPPORT SYSTEMS* 128, doi:10.1016/j.dss.2019.113157.

[10] Chamari I. Kithulgoda, Russel Pears & M. Asif Naeem (2018): *The incremental Fourier classifier: Leveraging the discrete Fourier transform for classifying high speed data streams*. *EXPERT SYSTEMS WITH APPLICATIONS* 97, pp. 1–17, doi:10.1016/j.eswa.2017.12.023.

[11] Tom Leinster (2016): *Basic Category Theory*. Available at `http://arxiv.org/abs/1612.09375`. Number: arXiv:1612.09375 arXiv:1612.09375 [math].

[12] Pierre-Francois Marteau (2021): *Random Partitioning Forest for Point-Wise and Collective Anomaly Detection-Application to Network Intrusion Detection*. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY* 16, pp. 2157–2172, doi:10.1109/TIFS.2021.3050605.

[13] Emily Riehl (2016): *Category Theory in Context*. Dover Publications Inc., Mineola, New York.

[14] Dan Shiebler (2022): *Kan Extensions in Data Science and Machine Learning*. arXiv:2203.09018.

[15] Dan Shiebler, Bruno Gavranović & Paul Wilson (2021): *Category Theory in Machine Learning*. *arXiv:2106.07032 [cs]*. arXiv:2106.07032.

[16] Sima Sobhiyeh & Mort Naraghi-Pour (2018): *Online Detection and Parameter Estimation with Correlated Data in Wireless Sensor Networks*. In: *2018 IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE (WCNC)*, IEEE Wireless Communications and Networking Conference, IEEE; IEEE Commun Soc; Natl Instruments; Rohde & Schwarz; Huawei; InterDigital; NEC. ISSN: 1525-3511.

[17] Sima Sobhiyeh & Mort Naraghi-Pour (2018): *Online hypothesis testing and non-parametric model estimation based on correlated observations*. In: *2018 IEEE GLOBAL COMMUNICATIONS CONFERENCE (GLOBECOM)*, IEEE Global Communications Conference, IEEE. ISSN: 2334-0983.

[18] Sin Yong Tan, Homagni Saha, Anthony R. Florita, Gregor P. Henze & Soumik Sarkar (2019): *A flexible framework for building occupancy detection using spatiotemporal pattern networks*. In: *2019 AMERICAN CONTROL CONFERENCE (ACC)*, Proceedings of the American Control Conference, Amer Automat Control Council; Int Federat Automat Control; Mitsubishi Elect Res Lab; Boeing; GE Res; United Technologies Res Ctr; General Motors Co; MathWorks; Halliburton; dSPACE; Int Journal Automat Comp; Altair; Soc Ind Appl Math; Wiley; Quanser; Temple Univ; Journal Franklin Inst; IEEE CAA Journal Automatica Sinica; Processes, pp. 5884–5889. ISSN: 0743-1619.

A Appendix

Model	Citation	AUC	Model	Citation	Acc (%)
EIF	[12]	0.9970	OEMP-4K	[17]	97
VAE	[12]	0.9960	BEMG	[17]	96.5
AD HKDE	[8]	0.9907	BEMP	[17]	96.5
C-SVM		0.9814	OEMP-2K	[17]	96.5
1C-SVM	[12]	0.9780	OEMP-3K	[17]	96.5
GRU-GSVDD	[4]	0.9109	Batch UM	[16]	96
GRU-GSVM	[4]	0.9059	Batch CBM	[16]	96
KDE-AB	[8]	0.9531	ARF	[10]	95.9
FOGD	[8]	0.9490	DA	[10]	95.8
IF	[12]	0.9470	AD	[10]	95.3
K-KDE	[8]	0.9368	OEMP-1K	[17]	95
Online osPCA	[8]	0.9292	RF	[9]	94.5
DIFF-RF	[12]	0.9000	J 48	[9]	94.4
LSTM-GSVM	[4]	0.8957	REPTree	[9]	94.4
GRU-QPSVM	[4]	0.8719	Occ-STPN	[18]	94
SVM	[4]	0.8676	**C-SVM**		93.5
LSTM-GSVDD	[4]	0.8609	IFC	[10]	93
CSVM	[3]	0.8220	HMM	[1]	90.2
DSVM	[3]	0.8220	LA	[10]	89.1
LSTM-QPSVM	[4]	0.8197	GeoMA	[17]	87
LSTM-QPSVDD	[4]	0.7869	LB	[10]	86.4
LSTM	[4]	0.7444	SOL	[10]	85.4
GRU-QPSVDD	[4]	0.7417	AS	[10]	84
SVDD	[4]	0.6715	PHT	[17]	83.5
LSVM	[3]	0.6670	OB	[10]	82.3
KitNET	[12]	0.6580	AUE	[10]	81.8
			RC	[10]	75.6
			AWE	[10]	73.4

Table 1: An ordered list of the ROCAUC scores (left) and accuracy scores (right) of unsupervised algorithms on the Occupancy Dataset as provided by papers found in Section **??**. Note that the number of decimal points is inconsistent between accuracy scores because different papers quoted the accuracy of their models to different precisions.