Pattern Recognition 101 (2020) 107198

Contents lists available at ScienceDirect

Pattern Recognition

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

# Towards explaining anomalies: A deep Taylor decomposition of one-class models

Jacob Kauffmann[a], Klaus-Robert Müller[a,b,c,*], Grégoire Montavon[a,*]

[a] Department of Electrical Engineering & Computer Science, Technische Universität Berlin, Marchstr. 23, Berlin 10587, Germany
[b] Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 02841, Korea
[c] Max Planck Institute for Informatics, Stuhlsatzenhausweg, Saarbrücken 66123, Germany

ARTICLE INFO

*Article history:*
Received 6 December 2018
Revised 3 December 2019
Accepted 8 January 2020
Available online 9 January 2020

*Keywords:*
Outlier detection
Explainable machine learning
Deep Taylor decomposition
Kernel machines
Unsupervised learning

ABSTRACT

Detecting anomalies in the data is a common machine learning task, with numerous applications in the sciences and industry. In practice, it is not always sufficient to reach high detection accuracy, one would also like to be able to understand *why* a given data point has been predicted to be anomalous. We propose a principled approach for one-class SVMs (OC-SVM), that draws on the novel insight that these models can be rewritten as distance/pooling neural networks. This 'neuralization' step lets us apply deep Taylor decomposition (DTD), a methodology that leverages the model structure in order to quickly and reliably explain decisions in terms of input features. The proposed method (called 'OC-DTD') is applicable to a number of common distance-based kernel functions, and it outperforms baselines such as sensitivity analysis, distance to nearest neighbor, or edge detection.



## 1. Introduction

Anomaly detection, or outlier detection, is a well-studied and well-formalized machine learning problem with numerous practical applications. One such application is intrusion detection in computer systems [11,14], where data points are e.g. messages transmitted over a network. Messages predicted as outliers are considered likely to carry a threat. The ability to detect outliers is also important in scientific applications, where points detected as such are intrinsically more interesting than inliers, and should therefore be given more attention [22,45]. A number of machine learning techniques have been successfully developed for outlier detection [32,33,39,44].

In practice, it is not only important to be able to detect outliers accurately, one also needs to understand why a machine learning model has reached a certain decision. An interpretable explanatory feedback can help the human to verify the decision making and potentially improve the model. Explaining machine learning decisions is especially important in applications where a potentially flawed ML model could have severe consequences in the real-world [6,9,18], or when aiming to extract novel scientific insight from a successfully trained ML model [1,15,19,34,36]. In the context of supervised learning, numerous methods for explanation have been proposed [3,4,20,25,29,37]. While some of them may be technically applicable (or extensible) beyond supervised learning, e.g. to anomaly detection, aspects of the anomaly explanation problem must be addressed specifically, in particular, *how* to quantify inlierness/outlierness, and *what* is the underlying structure of anomaly prediction.

We contribute by addressing these multiple facets of explaining anomalies, and we propose a practical solution for one-class SVMs (OC-SVM). In particular, we introduce a fairly general characterization of inlierness and outlierness in Euclidean space, and find that outlierness *nonlinearly* relates to the model's output. Furthermore, we find that inlierness/outlierness predictions can be rewritten as multilayer neural networks composed of radial basis functions and pooling layers. These 'neuralized' predictions can be subsequently explained by applying 'Deep Taylor Decomposition' [28] a method that leverages the neural network structure to quickly and systematically propagate the prediction to the input features.

The overall procedure which we call 'OC-DTD' is illustrated in Fig. 1. It applies to a number of common distance-based kernel functions (Gaussian, Laplacian, t-Student) and requires neither retraining nor access to the training data. The proposed method could potentially be adapted to other distance-based outlier models, e.g. [5,8,16,17,30].

* Corresponding authors.
*E-mail addresses:* j.kauffmann@tu-berlin.de (J. Kauffmann), klaus-robert.mueller@tu-berlin.de (K.-R. Müller), gregoire.montavon@tu-berlin.de (G. Montavon).

https://doi.org/10.1016/j.patcog.2020.107198

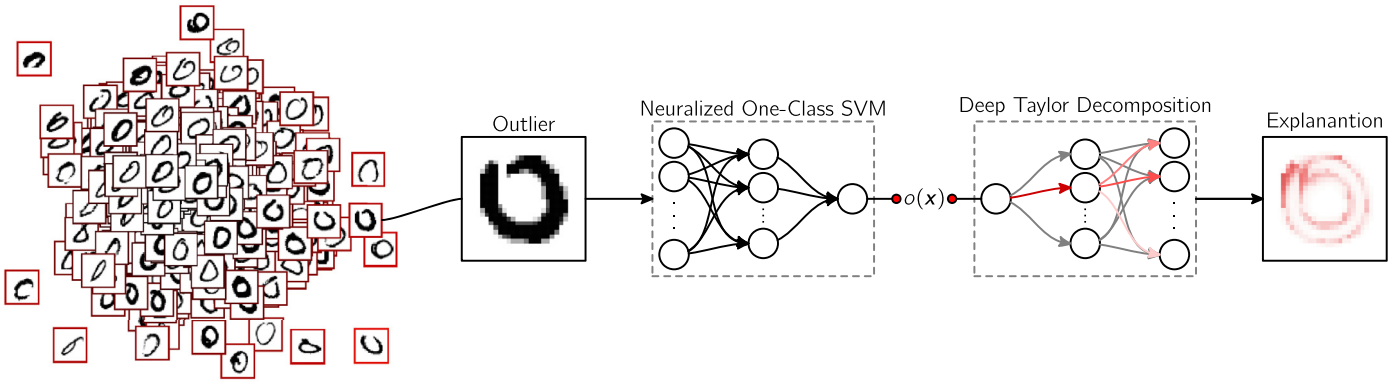


**Fig. 1.** Illustration of the proposed OC-DTD method for outlier explanation. We are given a one-class SVM that separates outliers from inliers. Predicted outlierness is first 'neuralized' in order to provide a structure for explanation. The neural network prediction for outlierness is then subject to a 'deep Taylor decomposition' which propagates the prediction backward in the network. The outcome is an attribution of the prediction on input features (visualized as a heatmap) indicating pixels that have contributed the most to outlierness.

Our experiments show that OC-DTD is able to identify meaningful input features responsible for outlierness, and does so in a way that outperforms baseline explanation techniques such as sensitivity analysis, difference to nearest neighbor, or edge detection.

*Related work*

Various methods have been proposed to explain machine learning decisions. Some methods explain by sampling or extracting the gradient near the analyzed data [29,37]. Other methods leverage the neural network structure of the model and explain by running a backward propagation pass in the network [3,43]. Further methods readily embed explanation structures into the model before training [9,46].

Some studies have focused on the problem of explaining outliers: Liu et al. [24] use the decision of a complex outlier detection model to train a set of simple detectors that separate outliers linearly from clusters of nearby training patterns. Subsequently, the linear weights are used to explain the outlier. Micenková et al. [26] remove features from detected outliers and return a subset of features that maximizes separability of the outlier from the surrounding training patterns. These two methods rely however on (1) the existence of a hypothetical outlier class that is approximated by sampling in the vicinity of the outliers and (2) access to the training data in the explanation stage. Schwenk and Bach [35] applied structured OC-SVMs to detect anomalies in MediaCloud applications, and proposed a technique to decompose predictions in terms of input variables for sum-decomposable kernels.

In this work we propose a general framework based on the novel concept of 'neuralization' which we combine with deep Taylor decomposition [28] in order to explain the predictions of a broad class of kernel one-class SVMs.

## 2. Quantifying inlierness and outlierness

In one-class learning, we are trying to separate patterns that are generated by one common distribution from the rest of the input domain. Schölkopf et al. [33] proposed the one-class SVM as an algorithm that learns the tails of a high-dimensional distribution, which is sufficient for the separation task. Technically, it projects the data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ in some feature space via a function $\Phi$ induced by some radial basis kernel function $\Bbbk$, and then learns a maximally separating hyperplane between the data and the origin. The optimization procedure results in a set of support vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m \in \mathbb{R}^d$ with $m \leq N$ together with positive coefficients $\alpha_1, \ldots, \alpha_m$ summing to 1 and a decision function:

$$g(\boldsymbol{x}) = \sum_{j=1}^{m} \alpha_j \, \Bbbk(\|\boldsymbol{x} - \mathbf{u}_j\|). \tag{1}$$

The function value is large for data points that are typical and small for anomalous data points. Other models such as Parzen window estimators or isotropic Gaussian mixture models can also be rewritten as such a one-class model.

To explain anomalies, a prerequisite is however to identify the quantity we would like to explain. In problems such as classification and regression, the output of the model can be readily interpreted, e.g. as the probability of membership to a given class, or as the expected value of the target variable respectively. When using a OC-SVM, such interpretation is not obvious: The discriminant function $g(\boldsymbol{x})$ does provide an ordering from the most to the least anomalous point (cf. [16]), however, it only answers which of two data points is most anomalous, and not the absolute level of anomaly of a given data point. Hence, we adopt an 'axiomatic approach', where we first give a set of necessary conditions for inlierness and outlierness, and then identify possible transformations of the function $g(\boldsymbol{x})$ that fulfill them.

**Definition 1.** A measure of inlierness $\mathbb{i} : \mathbb{R}^d \to \mathbb{R}$ must fulfill the following two conditions:

1. It is bounded by zero and some positive number $u$, i.e. $\forall \boldsymbol{x} : 0 \leq \mathbb{i}(\boldsymbol{x}) \leq u$.
2. It converges asymptotically to zero, i.e. $\forall \boldsymbol{x} \neq \mathbf{0} : \lim_{t \to \infty} \mathbb{i}(t\boldsymbol{x}) = 0$.

The upper-bound can be interpreted as the fact that a data point cannot be more inlier than the most prototypical example within the input distribution. The Gaussian mixture model, which is sometimes used for inlier/outlier detection (e.g. [38]), associates to each input point a score representing the likelihood of that point being generated from the modeled distribution. It is bounded between 0 and some constant, and converges to 0 when moving away from the data, thus fulfilling our definition of inlierness. Similarly, the discriminant function $g(\boldsymbol{x})$ of the OC-SVM is upper-bounded by the kernel bound, and converges to zero as we move away from the data. Thus, we can simply use

$$\mathbb{i}(\boldsymbol{x}) = g(\boldsymbol{x}) \tag{2}$$

as a measure of inlierness. We now consider the case of outlierness and give the following characterization of an outlier function:

**Definition 2.** A *measure of outlierness* $o : \mathbb{R}^d \to \mathbb{R}$ must fulfill the following two conditions:

1. It is lower bounded by zero, i.e. $\forall \boldsymbol{x}$: $0 \leq o(\boldsymbol{x})$,

2. It converges asymptotically with some predefined norm, i.e. $\forall \boldsymbol{x} \neq \mathbf{0} : \lim_{t\to\infty} o(t\boldsymbol{x})/\|t\boldsymbol{x}\|^q = c$, for some $q > 0$ and some $c > 0$.

This predefined norm corresponds to the natural metric of the input domain, and is typically assumed to be an $\ell_2$-norm in order to reflect the Euclidean geometry. Examples of functions that satisfy Definition 2 are the distance to the mean, or the negative log-likelihood (NLL) under an isotropic normal probability distribution. These functions are typically used in machine learning for measuring error. As a counterexample, the NLL of a general Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ learned from the data does not satisfy Definition 2: The latter is indeed not suitable for measuring outlierness, as the covariance $\Sigma$ overrides the natural metric of the input space on which the outlier decision should be based.

The quantity $g(\boldsymbol{x})$ is also clearly not a measure of outlierness: It asymptotes to 0 as $\boldsymbol{x}$ moves away from the data, which does not capture the fact that the degree of outlierness continues to increase. For a fairly broad class of kernels, however, measures of outlierness can be obtained by nonlinearly transforming $g(\boldsymbol{x})$. The first class of kernels that we consider are t-Student kernels $\mathbb{k}(\|\boldsymbol{x}-\boldsymbol{x}'\|) = (a + \|\boldsymbol{x}-\boldsymbol{x}'\|^q)^{-1}$, where $a > 0$ and $q \in \{1, 2, \ldots\}$. When the norm is scaled by a bandwidth, the kernel is also referred to as Cauchy kernel. A possible measure of outlierness is given by:

$$o(\boldsymbol{x}) = m \cdot g(\boldsymbol{x})^{-1}. \tag{3}$$

The second class of kernels we consider are exponential kernels $\mathbb{k}(\|\boldsymbol{x}-\boldsymbol{x}'\|) = \exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|^q/(q \cdot \sigma^q))$, where $\sigma > 0$ and $q \in \{1, 2, \ldots\}$. For $q = 1$ the kernel is called Laplacian, and for $q = 2$ Gaussian. A simple measure of outlierness is in this case:

$$o(\boldsymbol{x}) = -\log(g(\boldsymbol{x})). \tag{4}$$

A proof for the agreement of these outlier scores with Definition 2 can be found in Supplementary Note A.

## 3. Neuralizing one-class SVMs

Our goal is to explain inlierness and outlierness given by the functions $\mathbb{i}(\boldsymbol{x})$ and $o(\boldsymbol{x})$ in terms of input variables. In the context of deep neural networks, explanation methods have successfully exploited the structure of the neural network to quickly and stably produce explanations [3]. A priori, our one-class model does not have such neural network structure. Thus, a first step will be to convert the quantities to explain to neural networks. We refer to this process as 'neuralization'.

A two-layer neural network that computes *inlierness* is trivially obtained by dissociating the sum from the summands of Eq. (1):

**inlierness**

| | | |
|---|---|---|
| layer 1: | $h_j = \alpha_j \, \mathbb{k}(\|\boldsymbol{x} - \mathbf{u}_j\|)$ | (similarity) |
| layer 2: | $\mathbb{i} = \sum_{j=1}^{m} h_j$ | (max-pooling) |

The first layer computes the weighted similarities to the support vectors as measured by the kernel. The second layer performs a summing operation. Because the result of the sum is typically dominated by the largest terms, it can be interpreted as a soft max-pooling, that identifies whether the input is similar to *at least one* support vector.

To obtain a neural network structure for *outlierness*, one needs to consider the two classes of kernels separately. For the t-Student kernel, the measure of outlierness $o(\boldsymbol{x})$ can be implemented by the two-layer neural network:

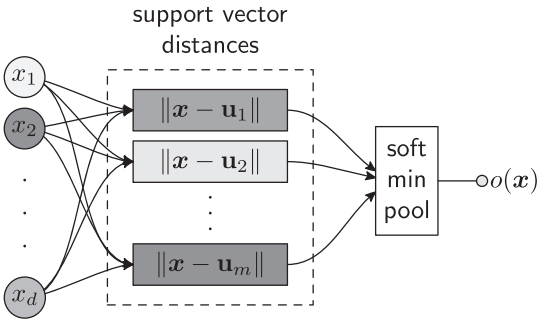


**Fig. 2.** Neuralized OC-SVM for computing outlierness (for inlierness and the sequential extension of outlierness, see Supplementary Note C).

**outlierness (t-Student kernel)**

| | | |
|---|---|---|
| layer 1: | $h_j = \frac{1}{\alpha_j} \cdot (a + \|\boldsymbol{x} - \mathbf{u}_j\|^q)$ | (distance) |
| layer 2: | $o = \mathrm{H}((h_j)_j)$ | (min-pooling) |

The first layer is a mapping to the effective distances $h_j$ from each support vector. By effective distance, we mean the distance as perceived by the data point $\boldsymbol{x}$, i.e. modulated by the support vector coefficients $\alpha_j$. The second layer computes the harmonic mean H which can be interpreted as a soft min-pooling. It verifies that the data point is distant from *all* support vectors. The neural network is illustrated in Fig. 2.

For the exponential kernels, of which the Laplacian and Gaussian kernels are special cases, the function $o(\boldsymbol{x})$ can be mapped to the two-layer neural network:

**outlierness (exponential kernel)**

| | | |
|---|---|---|
| layer 1: | $h_j = \frac{\|\boldsymbol{x} - \mathbf{u}_j\|^q}{q \cdot \sigma^q} - \log \alpha_j$ | (distance) |
| layer 2: | $o = -\mathrm{LSE}(-(h_j)_j)$ | (min-pooling) |

The network is structurally very similar to the t-Student case above, except that the soft min-pooling is this time implemented by a log-sum-exp (LSE) computation. Equivalence between one-class SVMs and the proposed neural networks is shown in Supplementary Note B.

When applied to sequential data such as images or text, one-class models based on RBF kernels become affected by the curse of dimensionality. Consequently, these models are often applied to small segments or patches $k$ of the input [13,23]. Inlier/outlier scores $(f_k)_k$ for all segments can then be pooled to compute a global score $f$ for the sequence:

**... extended to sequential data**

| | | |
|---|---|---|
| | $\vdots$ | |
| layer 3: | $f = \mathcal{P}((f_k)_k)$ | (pooling) |

The pooling function $\mathcal{P}$ is chosen such that it satisfies the definitions of Section 2. For example, a simple sum-pooling can be used when predicting outlierness. The neural network architecture for the sequential outlier model is also visualized in Supplementary Note C.

## 4. Explaining neural network predictions

Consider a given input example that our one-class SVM has predicted to be an inlier or an outlier. We study the problem of iden-

tifying which input features are most relevant for explaining the prediction. While this task is difficult for general nonlinear prediction functions, the preliminary "neuralization" step we have taken in Section 3 will make this process considerably easier. In particular, the problem of explanation will be decomposed into a large number of simple explanation tasks performed at each neuron. In other words, our "neuralized" one-class SVMs will serve as a backbone to guide the explanation process.

An explanation technique that leverages the neural network structure and was able to successfully explain complex models such as deep convolutional neural networks is layer-wise relevance propagation (LRP) [3]. LRP explains by taking the prediction as given by some output neuron and propagating it backwards in the neural network by means of purposely designed propagation rules. The procedure identifies relevant neurons at every layer and terminates once the input layer has been reached.

An important challenge for such explanation procedure is to make sure propagation rules perform meaningful redistribution to neurons in the layer below. This holds especially for novel architectures such as the proposed neuralized one-class SVMs, where intuition on how to set these propagation rules may be lacking.

### 4.1. Deep Taylor decomposition

Deep Taylor decomposition (DTD) [28] provides a principled framework to design LRP propagation rules in general neural network architectures. It performs for each neuron in the network a Taylor expansion of the quantity to be propagated. The linear terms of that expansion serve to determine how much relevance should be redistributed to each neuron in the lower layer.

Let $R_k$ be the relevance attributed to a certain neuron $k$ at a given layer. Let $\boldsymbol{h} = (h_j)_j$ denote neuron activations in the layer below. Deep Taylor decomposition views $R_k$ as a function of $\boldsymbol{h}$ and seeks to perform at some root point $\widetilde{\boldsymbol{h}}$ the Taylor expansion:

$$R_k(\boldsymbol{h}) = R_k(\widetilde{\boldsymbol{h}}) + \textstyle\sum_j [\nabla R_k(\widetilde{\boldsymbol{h}})]_j \cdot (h_j - \widetilde{h}_j) + \ldots$$

The linear term $[\nabla R_k(\widetilde{\boldsymbol{h}})]_j \cdot (h_j - \widetilde{h}_j)$ defines the share of $R_k$ that should be redistributed to neuron $j$ in the lower layer. The function $R_k(\boldsymbol{h})$ is however complex to analyze, especially after a few steps of propagation. A key aspect of DTD is thus to replace $R_k$ by a "relevance model" $\widehat{R}_k$ that uses only local information in the graph and that satisfies $\widehat{R}_k(\boldsymbol{h}) \approx R_k(\boldsymbol{h})$ at the current activations $\boldsymbol{h}$ and in its vicinity [28].

### 4.2. Propagation in pooling layers

Each model derived in Section 3 is characterized by the presence of a top-level pooling layer. Setting the top-layer relevance $R$ to the predicted quantity, we get one of the following functions:

$$R(\boldsymbol{h}) = \textstyle\sum_{j=1}^{m} h_j \qquad \text{(sum-pooling)}$$
$$R(\boldsymbol{h}) = \mathrm{H}((h_j)_j) \qquad \text{(harmonic pooling)}$$
$$R(\boldsymbol{h}) = -\mathrm{LSE}(-(h_j)_j) \qquad \text{(log-sum-exp pooling)}$$

Here, the functions are simple enough so that they can be analyzed directly, without having to build a relevance model. The next step is to compute linear terms $R_j = [\nabla R(\widetilde{\boldsymbol{h}})]_j \cdot (h_j - \widetilde{h}_j)$ representing the share received by each lower-layer neuron. In the following, we explore for each relevance function how to choose the root point $\widetilde{\boldsymbol{h}}$ and what redistribution rule we obtain as a result.

*Sum-pooling.* The sum-pooling was already treated in [28]: The sum-pooling function is linear and can therefore be represented exactly by a first-order Taylor expansion. When applied to positive activations, the only admissible root point is $\widetilde{\boldsymbol{h}} = \boldsymbol{0}$. Choosing this root point, linear terms of the Taylor expansion are trivially given by:

$$R_j = h_j$$

In the same way as the sum can be interpreted as a soft max-pooling, this operation can be interpreted as a 'max-take-most' redistribution.

*Harmonic pooling.* Although the harmonic pooling function is nonlinear, we observe that it is still linear on the segment $\{t \cdot \boldsymbol{h} | 0 < t \leq 1\}$. We then choose the root point $\widetilde{\boldsymbol{h}} = \lim_{t \to 0} t \cdot \boldsymbol{h}$ and perform a first order Taylor expansion at that point. Linear terms of the Taylor expansion are given by:

$$R_j = \frac{h_j^{-1}}{\sum_{j'} h_{j'}^{-1}} \cdot R \tag{5}$$

The redistribution is proportional to inverse activations, and can therefore be interpreted as a 'min-take-most' redistribution strategy.

*Log-sum-exp pooling.* Observing that the log-sum-exp pooling function is nonlinear, but linear on the line $\{\boldsymbol{h} - t \mid t \in \mathbb{R}\}$, we choose the root point $\widetilde{\boldsymbol{h}} = \boldsymbol{h} - o$, i.e. we subtract the output of the model to each dimension of the vector of activations. Linear terms of the Taylor expansion are this time given by:

$$R_j = \frac{\exp(-h_j)}{\sum_{j'} \exp(-h_{j'})} \cdot R \tag{6}$$

This propagation rule has a similar "min-take-most" interpretation as for the harmonic pooling case.

### 4.3. Propagation in RBF layers

The next step considers the function $R_j(\boldsymbol{x})$ and asks whether a Taylor expansion can be performed again in order to extract a propagation rule from the current layer to the input features. However, the function $R_j(\boldsymbol{x})$ is generally too complex to analyze directly. Hence, we will seek a relevance model $\widehat{R}_j(\boldsymbol{x})$ that accurately matches $R_j(\boldsymbol{x})$ locally, and that only depends on local quantities (e.g. the corresponding neuron activation).

**Proposition 1.** *Relevance scores obtained in Equation (5) can be rewritten as* $R_j = A_j h_j + B_j$ *with*

$$A_j = \frac{1}{m} \cdot \mathrm{H}\left(\left(\frac{h_{j'}}{h_j}\right)_{j'}\right)^2, \quad B_j = 0,$$

*and those of Equation (6) can be rewritten similarly with*

$$A_j = \frac{\exp\left(-h_j\right)}{\sum_{j'} \exp\left(-h_{j'}\right)}, \quad B_j = -A_j \cdot \mathrm{LSE}\Big(-\big(h_{j'} - h_j\big)_{j'}\Big).$$

*In both cases, the variables* $A_j$ *and* $B_j$ *converge to some constant value when support vector* $j$ *strongly dominates the min-pooling.*

See Supplementary Note D for a proof. As an illustration, Fig. 3 shows on a simple one-dimensional Gaussian kernel example with two support vectors, that the functions $R_j(\boldsymbol{x})$ and $h_j(\boldsymbol{x})$ start to coincide in regions of the input domain where $\mathbf{u}_j$ strongly dominates.

Motivated by Proposition 1, we build a relevance model $\widehat{R}_j$ that approximates $R_j$ as an affine function of $h_j$ where coefficients $A_j$ and $B_j$ are evaluated at the current data point and set constant. Observing that $h_j$ is itself also an affine function of kernel scores or power-raised distances, we build relevance models of the type:

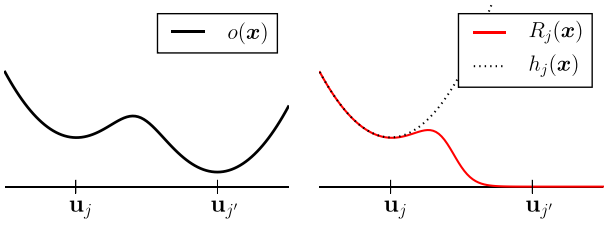


**Fig. 3.** Left: Outlier model. Right: Support vector relevance $R_j$ and activation $h_j$. The activation predicts the relevance well when $\mathbf{u}_j$ dominates locally.

$$\widehat{R}_j(\boldsymbol{x}) = C_j \cdot \mathbb{k}(\|\boldsymbol{x} - \mathbf{u}_j\|) + D_j \qquad \text{(kernel)}$$
$$\widehat{R}_j(\boldsymbol{x}) = C_j \cdot \|\boldsymbol{x} - \mathbf{u}_j\|^q + D_j \qquad \text{(distance)}$$

where the parameters $C_j$ and $D_j$ absorb the two successive affine transformations. The relevance models now depend in a simple manner on the input $\boldsymbol{x}$. We explore redistribution on the input features for the kernel and distance cases.

*Kernel.* This case occurs when explaining inlierness. Consider $\widehat{R}_j(\mathbf{u}_j)$, the maximum of the function. Although inlierness is strong at that location, the function looks the same in every direction, and it is therefore impossible to attribute this high score to a specific input direction. Only a more complex analysis involving e.g. pairs of support vectors would be able to provide meaningful directional information. In the proposed framework, an explanation of inlierness will thus be expressed in terms of support vectors $(\mathbf{u}_j)_j$ instead of being further redistributed to input dimensions.

*Distance.* This second case occurs when explaining outlierness. Here, $\widehat{R}_j(\boldsymbol{x})$ is an affine function of a power-raised distance function. When $q = 1$, we observe that the function $\widehat{R}_j$ is linear on the segment $\{\mathbf{u}_j + t \cdot (\boldsymbol{x} - \mathbf{u}_j) \mid 0 < t \leq 1\}$. We note that when $D_j < 0$ there is a root point $\widetilde{\boldsymbol{x}}_j$ on this segment, and we can use it as a reference point for the Taylor expansion. For $D_j \geq 0$, we choose as a reference point the function's minimum $\widetilde{\boldsymbol{x}}_j = \lim_{t\to 0} \mathbf{u}_j + t \cdot (\boldsymbol{x} - \mathbf{u}_j)$. A first-order Taylor expansion at the chosen reference point gives the redistribution messages:

$$R_{i\leftarrow j} = [\nabla \widehat{R}_j(\widetilde{\boldsymbol{x}}_j)]_i \cdot (x_i - \widetilde{x}_i^{(j)})$$

When $q \neq 1$, the first-order Taylor expansion used by DTD can be substituted by Integrated Gradients [37], where partial derivatives $\partial R_j / \partial x_i$ are integrated on the segment $[\widetilde{\boldsymbol{x}}_j, \boldsymbol{x}]$:

$$R_{i\leftarrow j} = \int_0^1 [\nabla \widehat{R}_j(\widetilde{\boldsymbol{x}}_j + t(\boldsymbol{x} - \widetilde{\boldsymbol{x}}_j))]_i \, dt \cdot (x_i - \widetilde{x}_i^{(j)})$$

The latter can be shown to be equivalent to messages obtained by Taylor expansion when $q = 1$. These messages are furthermore invariant to the choice of $q$, in particular, the propagation rule is always given by:

$$R_i = \sum_{j=1}^{m} \frac{(x_i - \mathrm{u}_i^{(j)})^2}{\|\boldsymbol{x} - \mathbf{u}_j\|^2} \cdot (R_j - D_j^+)$$

where we have sum-pooled messages $R_{i\leftarrow j}$ coming from all support vectors, and where $(\cdot)^+$ denotes the positive part. A detailed derivation is given in Supplementary Note E. This last step of propagation can be understood as a directional redistribution in input domain, modulated by the explainable part of the support vector relevance. Here, we observe that $\sum_{i=1}^{d} R_i = R - \sum_j D_j^+$, i.e. a certain fraction of the predicted outlierness cannot be attributed to the input variables.

The whole process of prediction and explanation is shown in Fig. 4. The data point (e.g. a handwritten digit) is given as input to the neuralized one-class SVM. Outlier scores are then redistributed using deep Taylor decomposition, first on the support vectors, and then propagated further on the input dimensions (e.g. pixels).

## 5. Implementing OC-DTD

Propagation equations in Section 4 are useful to interpret the explanation as a layer-wise redistribution process, however, these equations are not the most practical for implementation. We give here a restructuring of the overall OC-DTD computation that makes it implementable in a very compact and generalizable manner.

Let $\boldsymbol{d} = (\|\boldsymbol{x} - \mathbf{u}_j\|)_j$ be the vector of distances to the support vectors, $\boldsymbol{k} = (\mathbb{k}(\|\boldsymbol{x} - \mathbf{u}_j\|))_j$ the vector of kernel scores, and $\boldsymbol{\alpha} = (\alpha_j)_j$ the parameters of the one-class SVM. For inlierness, we compute

$$\boldsymbol{R} = \boldsymbol{\alpha} \odot \boldsymbol{k},$$

and for outlierness,

$$\boldsymbol{R} = \left(\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{x}}\right)^2 \cdot \left[\frac{\boldsymbol{d}}{q} \odot \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{d}} \odot \frac{\partial o}{\partial \boldsymbol{h}}\right], \qquad \text{(t-Student)}$$

$$\boldsymbol{R} = \left(\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{x}}\right)^2 \cdot \left[\min\left(o, \frac{\boldsymbol{d}}{q} \odot \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{d}}\right) \odot \frac{\partial o}{\partial \boldsymbol{h}}\right]. \qquad \text{(exponential)}$$

The squaring operation, the partial derivative $\partial \boldsymbol{h} / \partial \boldsymbol{d}$, and the min operation, all apply element-wise. A proof is given in Supplementary Note F. Details on how to exploit GPU implementations for efficient convolutional $\ell_2$ distance can be found in the appendix of [40]. Gradient computations can be easily implemented using *automatic differentiation*, a mechanism available in most neural network frameworks including PyTorch and TensorFlow.

## 6. Experiments

We first test OC-DTD on large images. We consider an outlier model over $7 \times 7$ image patches to which we apply the sequential extension proposed at the end of Section 3 with $o = \sum_k o_k$ as a top-level pooling function. We extract patches of the input image itself. From these patches, we build 1000 prototypes with $k$-means and then train a one-class SVM with a Gaussian kernel (exponential with $q = 2$) and the fraction of outliers $\nu = 0.1$ on the prototypes. We choose $\sigma$ in a way that 90% of the total of similarity scores under the kernel are contained in the 10% nearest neighbors. The fraction $\nu = 0.1$ outliers are then discarded and the OC-SVM is retrained on the inlier patches with a tighter "95% in 5%" ratio for $\sigma$ and with $\nu = 0.01$, essentially yielding a hard-margin OC-SVM on clean data.

Fig. 5 shows images taken from various datasets. Below each image is the corresponding OC-DTD pixel-wise explanation shown as a heatmap. We observe that heatmaps focus on outlier motives of respective images. They are also able to discard recurring patterns such as parking lines, grid structures, or grass. The third image (from the Describable Textures Dataset [10]) is tampered by rotating part of it by a very small angle. The rotation is hardly visible to the human eye, however the small artefacts that are introduced by the rotation are indeed discovered by OC-DTD, hence revealing the overall modification. The fourth image (taken from the VOC2007 dataset [12]) shows a more complex natural scene. OC-DTD identifies the rider's hat and the copyright tag as anomalous, while the background is completely discarded. Copyright tags were also identified in previous work as artefacts used by certain classification decisions [21].

Fig. 6 analyzes how OC-DTD performs at the pixel-level on the image of a ceiling, with a small defect in some dark area. The explanation is compared to the OC-SVM's patch-wise outlier scores $(o_k)_k$, a Sobel edge detector applied to the image, and an "edge $\times$ score" baseline consisting of multiplying the last two baselines. We can observe that OC-DTD detects the anomalous pattern more sharply and is also able to better filter out peripheral grid structures.

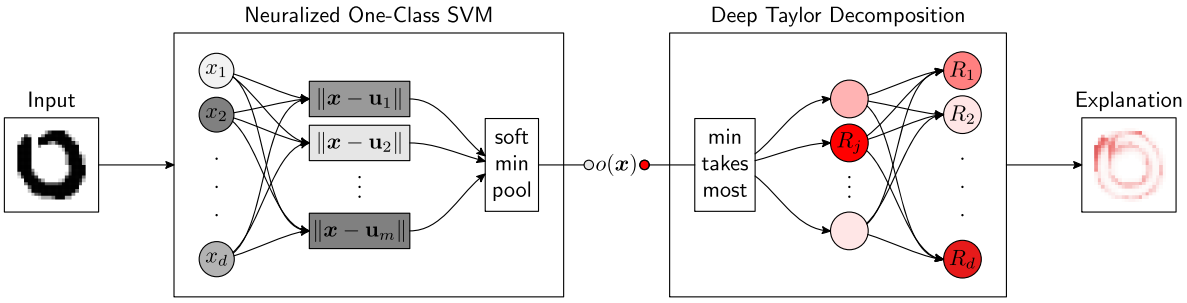


**Fig. 4.** Neural network equivalent of the OC-SVM for representing outlierness, and explanation by backward propagation from the output down to the input features. The explanation is visualized as a heatmap in pixel domain.

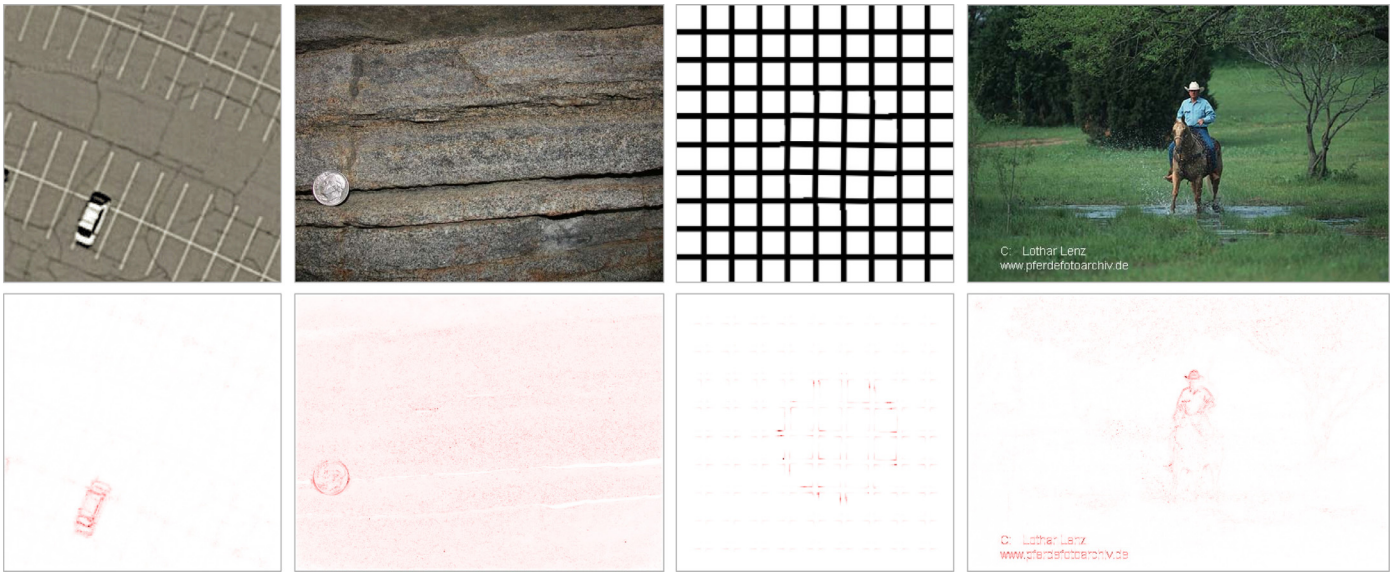

**Fig. 5.** Explanation of outlierness obtained by OC-DTD on large images. Pixels detected as anomalous are shown in red. Recurring patterns (e.g. parking lines, grass) are ignored and outlier elements (e.g. car, coin) are clearly highlighted. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

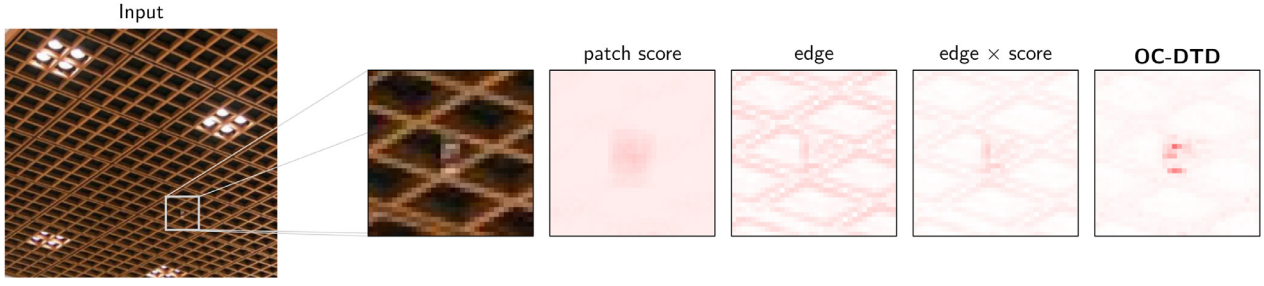


**Fig. 6.** Example of an image containing a small local artefact (zoomed-in), and explanations produced by some edge/score baselines and by OC-DTD. The OC-DTD method is more focused on the outlier pattern than the baselines.

### 6.1. External validation

The following experiments validate the ability of OC-DTD to correctly identify patterns of outlierness in the pixel-domain on an artificial problem where we have ground truth information.

We take the *original* MNIST test set, and build a *modified* version of it where a stroke artefact is superposed to each example. The artefact is generated by sampling three points: one on the left edge, one in the center and one on the right edge of the image and with random vertical coordinates. The three points are then connected with a Bézier curve. Examples are given in Fig. 7 (first row).

A one-class SVM with a Gaussian kernel is trained on the MNIST training set. Considering the original test set as "inliers" and the modified set as "outliers" gives an area under the ROC curve (AUROC) of 0.952, indicating that the model has fairly well identified the stroke artefact as outlier.

Explanations for the outlierness of the original and modified test set are then produced with OC-DTD. Fig. 7 (second row) shows the result for a few examples. Explanations uniformly and consistently highlight the artificially added stroke artefact as a source of outlierness. The explanations also highlight the top-right corner of digit 2, where we can clearly see an artefact. We also observe a residual edge effect present in each digit. It can be attributed to small disalignments between the digit and the nearby support vectors, which are then carried by OC-DTD into the explanation.

We then compare OC-DTD explanations to another type of explanation derived from a multivariate normal density model (MVN), which is *not* an outlier model in the sense of Definition 2. The negative log-likelihood function implemented by this model is

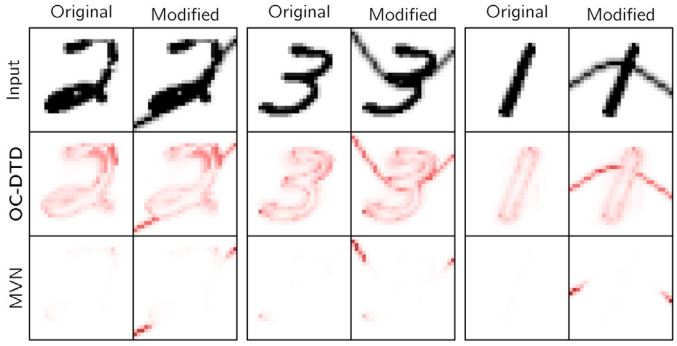


**Fig. 7.** *Top row:* Randomly selected examples from the MNIST test set shown next to their modified version. *Bottom rows:* pixel-wise outlierness found by OC-DTD and a multivariate normal density model (MVN).

given by $\mathrm{NLL}(\boldsymbol{x}) = \mathrm{NLL}(\boldsymbol{\mu}) + \sum_{i=1}^{d}(x_i - \mu_i)^2/2\sigma_i^2$ where $\boldsymbol{\mu}$ and $(\sigma_i)_i$ are maximum likelihood mean vector and scale parameters estimated from the data, and where off-diagonal covariances are fixed to zero. After training, we obtain an AUROC of 1.000, indicating that the MVN model of outlierness is very responsive to the stroke artefact (even more than the one-class SVM).

The MVN prediction can be explained by identifying the summands of the negative log-likelihood as the contribution of input variables, i.e. $R_i = (x_i - \mu_i)^2/2\sigma_i^2$. Fig. 7 (third row) shows the resulting explanations. Unlike the OC-DTD explanations, the MVN model puts a much stronger emphasis on part of the strokes that are near the border, and ignoring artefactual elements that are closer to the actual digit. The reason for the predominance of stroke artefacts in the MVN explanation is that the learned variance parameters distort the natural metric of the input domain.

Denoting the original and modified digit as $\boldsymbol{x}$ and $\boldsymbol{x}'$, our observations can be verified quantitatively by computing the cosine similarity between the artefact pattern $\boldsymbol{x}' - \boldsymbol{x}$ and its effect on explanation $R(\boldsymbol{x}') - R(\boldsymbol{x})$. Cosine similarity scores are then averaged over the MNIST test set. A score close to 1.0 indicates that the detected artefacts are attributed to the correct pixels. OC-DTD reaches an average cosine similarity of 0.87, largely superior to the MVN, which reaches a cosine similarity of 0.58 only.

### 6.2. Internal validation

In the following experiments, we consider the output of the one-class SVM to be our ground-truth measure of outlierness. The OC-DTD method for explanation will be compared to a number of baseline explanation techniques that we detail in Section 6.2.2.

We train a one-class SVM on the original MNIST images, and on 30000 randomly selected patches of size $7 \times 7$ from the CIFAR-10 dataset. Pixels are coded between $-1$ and $+1$. At test time the outlier scores of CIFAR-10 examples are obtained by summing over all patches of an image. For the t-Student kernels, we choose $a = q\sigma^q$, which mimics the shape of the exponential kernel for small distances, but has heavier tails.

#### 6.2.1. Pixel-flipping evaluation metric

For evaluation of explanation quality, we consider the pixel-flipping approach [31] that was proposed in the context of image classifiers. The approach consists of gradually destroying pixels from most to least relevant according to the explanation, and measuring how quickly the prediction score decreases.

In the context of outlier detection, however, destroying a pixel does not reduce evidence for outlierness and might even create more of it. Thus, the original pixel-flipping method must be adapted to the specific outlier detection problem. Our approach will consist of performing the flipping procedure not in the pixel-space directly, but in some feature space

$$\Psi(\boldsymbol{x}) = \begin{bmatrix} x_1 - \mathrm{u}_1^{(1)} & \dots & x_1 - \mathrm{u}_1^{(m)} \\ \vdots & \ddots & \vdots \\ x_d - \mathrm{u}_d^{(1)} & \dots & x_d - \mathrm{u}_d^{(m)} \end{bmatrix}.$$

containing all component-wise differences to support vectors. The OC-SVM can be rewritten in terms of elements of this feature space as $g(\Psi) = \sum_j \alpha_j \Bbbk(\|\Psi_{:,j}\|)$, and similarly the outlier function can be written as $o(\Psi)$.

In our modified pixel-flipping procedure, each flip corresponds to setting one row of $\Psi$ (the one with most assigned relevance) to zero. This procedure can be interpreted as progressively marginalizing the relevant dimensions. Once dimensionality 0 is reached, the pattern is necessarily an inlier, because no deviation from the support vectors exists anymore. The pixel-flipping procedure is detailed in Algorithm 1.

**Algorithm 1** Pixel-flipping procedure.

**inputs**
  $\Psi(\boldsymbol{x})$ ▷ Effective inputs
  $\mathbf{R}$ ▷ Heatmap
**outputs**
  `pfcurve` ▷ Declining outlier score
**procedure**
  `pfcurve` ← `[]`
  **for** `i` **in** `argsort`$(-\mathbf{R})$ **do**
    $\Psi_{\mathtt{i},:} \leftarrow \mathbf{0}$
    `pfcurve.append`$(o(\Psi))$

Pixel-flipping curves are computed for all examples in the test set and then averaged to create a mean pixel-flipping curve.

#### 6.2.2. Baselines

We compare OC-DTD to a number of baselines for explanation, that we group in the following categories:

*Distance Decomposition.* Our first baseline computes $(\boldsymbol{x} - \mathbf{u}_{\mathrm{NN}})^2$ the element-wise squared difference to the nearest support vector. This method can be seen as a variant of OC-DTD where the min-take-most redistribution step is substituted by min-take-all. This results in strongly localized explanations, but also causes discontinuity when $\boldsymbol{x}$ transitions from one nearest neighbor to another. A second baseline computes $(\boldsymbol{x} - \bar{\mathbf{u}})^2$ the squared difference to the average of support vector $\bar{\mathbf{u}} = \frac{1}{m}\sum_{j=1}^{m}\mathbf{u}_j$.

*Gradient-Based.* These baselines involve the local evaluation of the gradient: the gradient itself ($\nabla o(\boldsymbol{x})$), the gradient multiplied by the input ($\nabla o(\boldsymbol{x}) \odot \boldsymbol{x}$) and sensitivity analysis ($\nabla o(\boldsymbol{x})$)$^2$. The last baseline can be interpreted as assigning high relevance to input features whose local perturbation causes a high variation of outlierness. A further extension (cf. [37]) integrates the gradient $\nabla o$ on a segment connecting some reference point $\tilde{\boldsymbol{x}}$ and the data point $\boldsymbol{x}$. A first variant of integrated gradients chooses $\tilde{\boldsymbol{x}} = \mathbf{u}_{\mathrm{NN}}$ the nearest support vector, and a second variant chooses $\tilde{\boldsymbol{x}} = \bar{\mathbf{u}}$ the mean of the support vectors.

*SHAP.* This method (cf. [25]) explains based on a sampling approximation of Shapley values. In our experiments, we use the publicly available kernel SHAP implementation [25]. On MNIST, random pixels (px) or image segments (seg) are flipped towards a randomly chosen $k$-means centroid ($k = 50$) and a weighted linear model is trained on 500 images with 50% pixels/segments set to zero on average. On CIFAR-10, SHAP did not produce competitive results.

*Edge Detection.* We consider the output of the Sobel edge detector. For the sequential model used on CIFAR-10, the output of

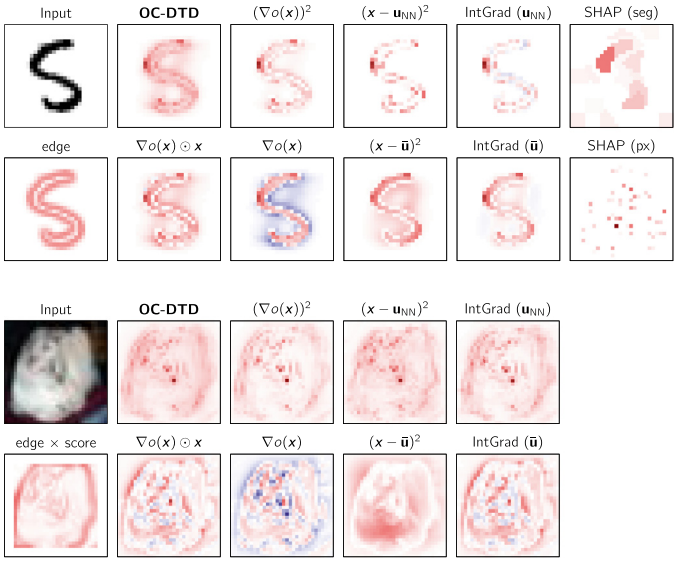


**Fig. 8.** Explanations produced by OC-DTD and various baselines on the OC-SVM trained on MNIST (top) and the patch-based OC-SVM trained on CIFAR-10 (bottom). A consistent feature of OC-DTD is that heatmaps are denser than the baselines.

the edge detector is multiplied by patch-wise outlier scores zero-padded to the original image.

### 6.2.3. Results

We consider the problem of explaining OC-SVMs with exponential kernel and $q = 2$ (i.e. Gaussian kernel). Examples of explanations produced by OC-DTD and the baselines are given in Fig. 8. At the exception of few methods such as edge detection or simple gradient, most methods agree on the overall pixels that are responsible for outlierness. A distinctive feature of OC-DTD is that the explanations are denser and distribute outlierness to more pixels than for other methods.

Sensitivity analysis is similar to OC-DTD, but a bit sparser. On MNIST, methods based on nearest neighbor $\mathbf{u}_{NN}$ produce even sparser explanations: Any pixel that is identical between the data point and the nearest neighbor will necessarily be assigned zero relevance although it may still have contributed to outlierness. Difference to the mean $(\boldsymbol{x} - \bar{\mathbf{u}})^2$ produces more comprehensive explanations, however, it less capable of modeling anomalies on CIFAR-10, where white and black image patches are systematically over-represented in the explanation.

Other baselines are subject to diverse specific limitations. Edge detection cannot highlight anomalous flat surfaces as relevant. Simple gradient ($\nabla o(\boldsymbol{x})$) cannot highlight background pixels as positively relevant, and SHAP shows a tradeoff between spatial resolution and comprehensiveness of the explanation.

Fig. 9 (top) shows the results of the pixel-flipping experiment for all explanation methods, using the same OC-SVMs as above. A number of methods are competitive, especially on MNIST, thus, the plot below provides a zoomed version of these curves relative to the OC-DTD pixel-flipping curve.

On MNIST, no method consistently dominates through the whole pixel-flipping procedure. OC-DTD is on average competitive, but it is dominated by nearest neighbor and edge detector methods for few pixel flips, and mean-based methods for a larger number of pixel flips. On CIFAR-10, OC-DTD is consistently better than all baselines. The curiously high performance of the edge × score baseline at $n = 676$ is due to the score zero-padding effect.

Pixel-flipping results are summarized in Fig. 9 (bottom) by ranking the methods according to their score at each stage of the pixel-flipping procedure, and taking the average rank over the different stages. We can observe that OC-DTD ranks lower than the baselines on the two datasets and for all choices of kernel.

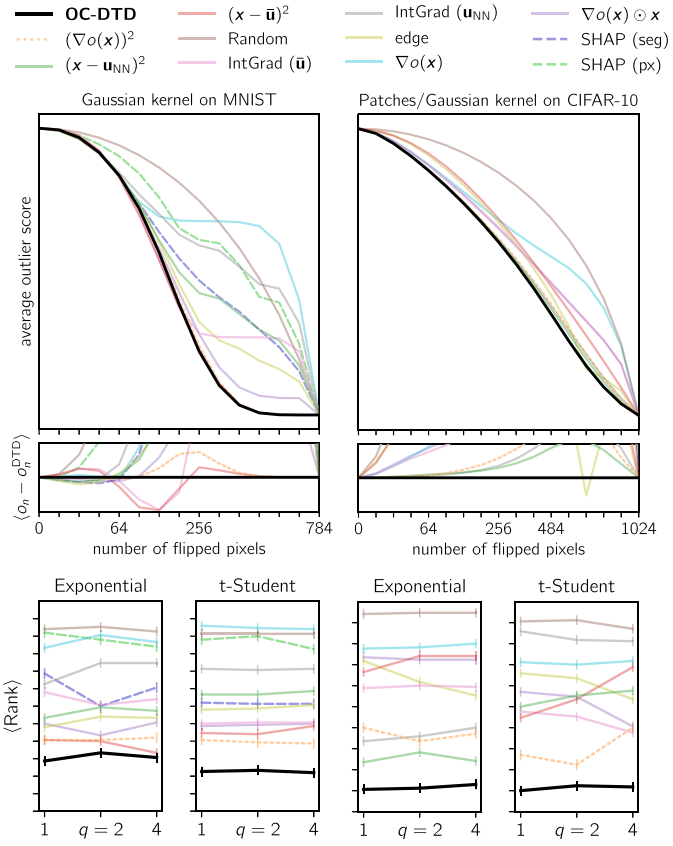


**Fig. 9.** *Top:* Pixel-flipping curves of various explanation techniques for the OC-SVM models trained on MNIST and CIFAR-10. The lower the curve, the better the explanations. *Bottom:* Average rank metric for various choices of kernels. The lower the better.

To determine whether our proposed OC-DTD method is better for all data points or only on average, we plot in Fig. 10 (left) pixel-flipping AUC scores[1] of OC-DTD versus some competitive baselines, for individual data points. We observe that OC-DTD performs consistently better than sensitivity analysis and dominates all methods in presence of mild anomalies (AUC low). The baseline $(\boldsymbol{x} - \bar{\mathbf{u}})^2$ performs better on MNIST for strong anomalies (AUC large), however, we note that this is also where the pixel-flipping metric is the least reliable due to the difficulty of performing meaningful digit reconstruction.

To get further insight on when this difference of performance is the strongest or weakest, we highlight a few extremal examples in the scatter plots, and visualize their explanations. We observe that when CIFAR-10 images consist of well-identifiable objects (e.g. a plane on some uniform background), sensitivity analysis performs equally well or even better than OC-DTD. Sensitivity analysis performs however poorly on images that have a more complex structure. This lower performance of sensitivity analysis can be traced down to the propagation of signed gradients, where gradients of adjacent patches risk to be canceled.

## 6.3. Intrusion detection

One-class SVM has been applied to network intrusion detection and malware detection [14,41]. Having interpretable model outputs

[1] Here, we use the AUC metric instead of the rank in order to produce continuous values in the scatter plot.

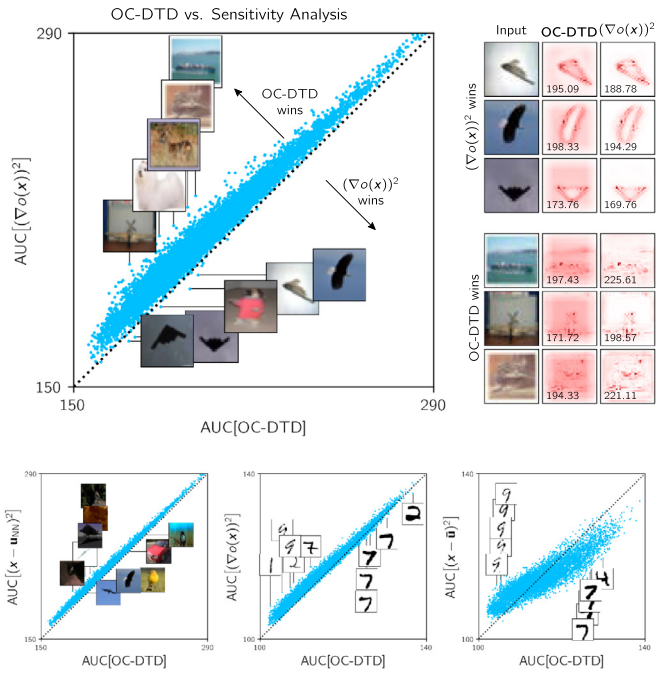


**Fig. 10.** *Top left:* Scatter plot comparing pixel-flipping AUC performance of OC-DTD and a baseline method on a single example basis on CIFAR-10. *Top right:* Heatmaps for cases where OC-DTD performed best or worst compared to sensitivity analysis. AUC values are given at the bottom left of every heatmap. *Bottom left to right:* $(\boldsymbol{x} - \mathbf{u}_{\text{NN}})^2$ baseline on CIFAR-10, sensitivity analysis on MNIST, $(\boldsymbol{x} - \bar{\mathbf{u}})^2$ baseline on MNIST.

::Fantastic! Another idiot who can't spell. You mods really stick together, don't you? That's so sweet. *e-tear*

**Most relevant terms in document:** fantast, idiot, mod, sweet, tear, stick, spell, togeth, anoth, realli,
**Most relevant terms not in document:** the, to, and, of, is, in, it, for, be, this,

Be quit, stop using poor excuses with the above links, i included a very credible link, you do not want it there, becaue you did not check on it in the first place, which was authentic you know. [...]

**Most relevant terms in document:** toilet, gordon, authent, excus, credibl, poor, remain, deal, stop, kind,
**Most relevant terms not in document:** and, this, as, if, or, articl, an, at, about, by,

**Fig. 11.** Word-level relevance assignment of OC-DTD for two sample message from the Detox dataset (the second message is truncated). Red color indicates high relevance scores.

can help to identify the intent or the method of an attack. We take up this idea in a simpler setting where no domain knowledge is necessary and where it is arguably possible to detect outlierness on a symbolic level, where it can be related to an attack. In particular, we train a OC-SVM on the personal attacks corpus from the Detox dataset [42]. In this dataset, documents are labeled by up to ten annotators as either 0 (neutral) or 1 (personal attack).

A dictionary is constructed from stemmed terms that appear in at least five documents and binary features are extracted as a vectorial representation of documents. No stop words are removed and no document frequencies are used for feature extraction. The model is trained on neutral examples (those with label mean 0). We use a Gaussian kernel and set $\nu = 0.3$. Parameter $\sigma$ is set to 10, which is a soft assumption of an expected difference in 10 terms for similar documents.

Fig. 11 shows the explanation for two sample messages, where relevant words are highlighted [2]. As one would expect, terms associated to a personal attack are identified as relevant. Note that some words such as 'fantastic' and 'authentic' are highlighted, not because they represent an attack, but simply because the are less frequently used in standard text. Conversely, common terms have no or low relevance in the document. Due to the RBF property, relevance will also be assigned to terms that do *not* appear in a document. These terms can be interpreted as being typical and their absence is therefore considered anomalous.

## 7. Conclusion

In this paper, we have addressed the question of explaining anomalies, by proposing a deep Taylor decomposition of the one-class SVM. The method is applicable to a number of commonly used kernels, and produces explanations in terms of support vectors or input variables. Our empirical analysis has demonstrated that the proposed method is able to reliably explain a wide range of outliers, and that these explanations are more robust than those obtained by methods such as sensitivity analysis or nearest neighbor.

A crucial aspect of our explanation method is the construction of 'neuralized' versions of the one-class SVM, that serve as a backbone to guide the process of explanation. These neural networks have in turn highlighted the asymmetry between the problem of inlier and outlier detection, where the first one can be modeled as a max-pooling over similarities, and where the latter is better modeled as a min-pooling over distances.

These neural networks may also be useful beyond the task of explanation, for example, to analyze layer after layer [7,27] the forming of outlierness. Furthermore, the novel insight on the structure of the outlier detection problem could inspire in the future the design of deeper and more structured outlier detection models.

## Declaration of Competing Interest

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

## Acknowledgments

This research was supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2017-0-00451, No. 2017-0-01779); the Deutsche Forschungsgemeinschaft (DFG) [grant MU 987/17-1]; the German Ministry for Education and Research as Berlin Big Data Center (BBDC) [01IS14013A] and Berlin Center for Machine Learning (BZML) [01IS18037A]. We are grateful to Guido Schwenk for the valuable discussion.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2020.107198.

## References

[1] B. Alipanahi, A. Delong, M.T. Weirauch, B.J. Frey, Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning, Nat. Biotechnol. 33 (8) (2015) 831–838.
[2] L. Arras, F. Horn, G. Montavon, K.-R. Müller, W. Samek, "What is relevant in a text document?": an interpretable machine learning approach, PLoS ONE 12 (8) (2017) 1–23.
[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS ONE 10 (7) (2015) 1–46.
[4] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K.-R. Müller, How to explain individual classification decisions, J. Mach. Learn. Res. 11 (2010) 1803–1831.
[5] C.M. Bishop, Novelty detection and neural network validation, IEE Proc.-Vis. Image Signal Process. 141 (4) (1994) 217–222.

[6] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L.J. Ackel, U. Muller, P. Yeres, K. Zieba, VisualBackProp: efficient visualization of CNNs for autonomous driving, in: IEEE International Conference on Robotics and Automation, 2018, pp. 1–8.
[7] M.L. Braun, J.M. Buhmann, K.-R. Müller, On relevant dimensions in kernel feature spaces, J. Mach. Learn. Res. 9 (2008) 1875–1908.
[8] M.M. Breunig, H. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.
[9] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, N. Elhadad, Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission, in: International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1721–1730.
[10] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.
[11] D.E. Denning, An intrusion-detection model, IEEE Trans. Softw. Eng. 13 (2) (1987) 222–232.
[12] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.
[13] A. Frome, Y. Singer, F. Sha, J. Malik, Learning globally-consistent local distance functions for shape-based image retrieval and classification, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.
[14] N. Görnitz, M. Kloft, K. Rieck, U. Brefeld, Toward supervised anomaly detection, J. Artif. Intell. Res. 46 (2013) 235–262.
[15] K. Hansen, D. Baehrens, T. Schroeter, M. Rupp, K.-R. Müller, Visual interpretation of kernel-based prediction models, Mol Inform 30 (9) (2011) 817–826.
[16] S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, K.-R. Müller, From outliers to prototypes: ordering data, Neurocomputing 69 (13) (2006) 1608–1618.
[17] H. Hoffmann, Kernel PCA for novelty detection, Pattern Recognit. 40 (3) (2007) 863–874.
[18] D. Kamarinou, C. Millard, J. Singh, Machine learning with personal data, Queen Mary School Law Legal Stud. Res.Paper 247 (2016).
[19] O.Z. Kraus, L.J. Ba, B.J. Frey, Classifying and segmenting microscopy images with deep multiple instance learning, Bioinformatics 32 (12) (2016) 52–59.
[20] W. Landecker, M.D. Thomure, L.M.A. Bettencourt, M. Mitchell, G.T. Kenyon, S.P. Brumby, Interpreting individual classifications of hierarchical networks, in: IEEE Symposium on Computational Intelligence and Data Mining, 2013, pp. 32–38.
[21] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking Clever Hans predictors and assessing what machines really learn, Nat. Commun. 10 (2019) 1096.
[22] J. Laurikkala, M. Juhola, E. Kentala, Informal identification of outliers in medical data, The Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology, 2000.
[23] F. Li, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 524–531.
[24] N. Liu, D. Shin, X. Hu, Contextual outlier interpretation, in: International Joint Conference on Artificial Intelligence, 2018, pp. 2461–2467.
[25] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2017, pp. 4765–4774.
[26] B. Micenková, X.-H. Dang, I. Assent, R.T. Ng, Explaining outliers by subspace separability, in: IEEE International Conference on Data Mining, 2013, pp. 518–527.
[27] G. Montavon, M.L. Braun, K.-R. Müller, Kernel analysis of deep networks, J. Mach. Learn. Res. 12 (2011) 2563–2581.
[28] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, Explaining nonlinear classification decisions with deep Taylor decomposition, Pattern Recognit. 65 (2017) 211–222.
[29] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: ACM International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[30] L. Ruff, N. Görnitz, L. Deecke, S.A. Siddiqui, R.A. Vandermeulen, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: Proceedings of the 35th International Conference on Machine Learning, 2018, pp. 4390–4399.
[31] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, Evaluating the visualization of what a deep neural network has learned, IEEE Trans. Neural Netw. Learn. Syst. 28 (11) (2017) 2660–2673.
[32] T. Schlegl, P. Seeböck, S.M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: International Conference on Information Processing in Medical Imaging, 2017, pp. 146–157.
[33] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, J.C. Platt, Support vector method for novelty detection, in: Advances in Neural Information Processing Systems, 1999, pp. 582–588.
[34] K.T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, A. Tkatchenko, Quantum–chemical insights from deep tensor neural networks, Nat. Commun. 8 (2017) 13890.
[35] G. Schwenk, S. Bach, Detecting behavioral and structural anomalies in MediaCloud applications, CoRR abs/1409.8035 (2014).
[36] I. Sturm, S. Bach, W. Samek, K.-R. Müller, Interpretable deep neural networks for single-trial EEG classification, J. Neurosci. Method. 274 (2016) 141–145.
[37] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: International Conference on Machine Learning, 2017, pp. 3319–3328.
[38] D.M.J. Tax, R.P.W. Duin, Outlier detection using classifier instability, in: Advances in Pattern Recognition, Joint IAPR International Workshops, 1998, pp. 593–601.
[39] D.M.J. Tax, R.P.W. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.
[40] M. van der Wilk, C.E. Rasmussen, J. Hensman, Convolutional gaussian processes, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, 2017, pp. 2849–2858.
[41] Y. Wang, J. Wong, A. Miner, Anomaly intrusion detection using one class SVM, in: IEEE SMC Information Assurance Workshop, 2004, pp. 358–364.
[42] E. Wulczyn, N. Thain, L. Dixon, Ex machina: personal attacks seen at scale, in: International Conference on World Wide Web, 2017, pp. 1391–1399.
[43] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, 2014, pp. 818–833.
[44] S. Zhai, Y. Cheng, W. Lu, Z. Zhang, Deep structured energy based models for anomaly detection, in: International Conference on Machine Learning, 2016, pp. 1100–1109.
[45] Y.-X. Zhang, A.-L. Luo, Y.-H. Zhao, Outlier detection in astronomical data, Optimizing Scientific Return for Astronomy through Information Technologies, SPIE, 2004.
[46] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

**Jacob Kauffmann** received a Bachelors degree in Computer Science from TU Berlin in 2014 and a Masters degree in Computer Science from TU Berlin in 2017. He is currently a Ph. D. student in the Machine Learning Group at TU Berlin.

**Klaus-Robert Müller** (Ph.D. 92) has been a Professor of computer science at TU Berlin since 2006; co-director Berlin Big Data Center. He won the 1999 Olympus Prize of German Pattern Recognition Society, the 2006 SEL Alcatel Communication Award, and the 2014 Science Prize of Berlin. Since 2012, he is an elected member of the German National Academy of Sciences Leopoldina.

**Grégoire Montavon** received a Masters degree in Communication Systems from École Polytechnique Fédérale de Lausanne, in 2009 and a Ph.D. degree in Machine Learning from the Technische Universität Berlin, in 2013. He is currently a Research Associate in the Machine Learning Group at TU Berlin.