# Fusing Dictionary Learning and Support Vector Machines for Unsupervised Anomaly Detection

Paul Irofti[a], Andrei Hîji[a], Andrei Pătrașcu[a], Nicolae Cleju[b]

[a] *Research Center for Logic, Optimization and Security (LOS), Department of Computer Science, Faculty of Mathematics and Computer Science, University of Bucharest, Romania*
[b] *Faculty of Electronics, Telecommunications and Information Technology Technical University of Iași, Romania*

**Abstract**

We study in this paper the improvement of one-class support vector machines (OC-SVM) through sparse representation techniques for unsupervised anomaly detection. As Dictionary Learning (DL) became recently a common analysis technique that reveals hidden sparse patterns of data, our approach uses this insight to endow unsupervised detection with more control on pattern finding and dimensions. We introduce a new anomaly detection model that unifies the OC-SVM and DL residual functions into a single composite objective, subsequently solved through K-SVD-type iterative algorithms. A closed-form of the alternating K-SVD iteration is explicitly derived for the new composite model and practical implementable schemes are discussed. The standard DL model is adapted for the Dictionary Pair Learning (DPL) context, where the usual sparsity constraints are naturally eliminated. Finally, we extend both objectives to the more general setting that allows the use of kernel functions. The empirical convergence properties of the resulting algorithms are provided and an in-depth analysis of their parametrization is performed while also demonstrating their numerical performance in comparison with existing methods.

*Keywords:* anomaly detection, dictionary learning, one-class support vector machines, kernel methods, sparse representations

## 1. Introduction

The anomaly detection task seeks to find the needle in the haystack [1]. Given a large dataset its aim is to identify the few particular points, called anomalies or outliers, that stand out against the commonality of the vast majority (also called inliers). The two classes are highly unbalanced and the ratio of outliers, called contamination rate, can often be below one percent. We are particularly interested in the unsupervised setting where we are given an unlabeled dataset based on which our machine learning algorithms have to separate the two classes [2]. In order to achieve this goal, in this paper we study the possibility of connecting the objective of two well established methods in the field: dictionary learning [3] and one-class support vector machines [4]. We are motivated by the promising results obtained in the literature when fusing the two methods in the supervised learning setting [5, 6, 7].

In real-world applications such as industrial sensor networks, cybersecurity, and medical screening, we rarely, if ever, have access to labeled data leading to an increased demand for reliable unsupervised anomaly detection algorithms [2]. Unlike supervised approaches, unsupervised methods scale without the need for expensive annotations, can adapt to new data and avoid dataset biases making them a good fit for early and zero-day detection. There is extensive work supporting this claim for various tasks and settings such as robustness towards new outlier types [8], subspace feature selection [9], and contrastive learning in graphs [10]. Besides the active interest in the field, our motivation is also driven by the fact that our proposed unsupervised anomaly detection method fusing dictionary learning and one-class support vector machines has little coverage in existing literature.

Dictionary learning (DL) [3] is a matrix factorization method where a redundant base, called dictionary or frame, is learned such that it provides sparse representations [11] for the given dataset. Nowadays, dictionary learning is a well established standard machine learning method with multiple extensions, variants and specializations that lead to various applications across fields such as audio and image processing [12], compression[13], timeseries [14], fault detection and isolation [15], computer vision [16] and classification [17].

Let the data items in the given dataset be organized as columns inside a large matrix. The main DL strength over similar dimensionality-reduction techniques, such as principal component analysis or low-rank approximations, is its ability to find a separate sub-space for each data-item in the dataset. In other words, the resulting representation columns are sparse and their support is not similar. In fact it was recently shown that they follow a distribution mixture composed of multiple multivariate variables following a truncated normal distribution [18].

Starting from the column support distribution described above, we are now interested in selecting only a few lines with whom we can still adequately represent the inliers but not the outliers. The support for each data-item remains sparse and distinct from the other columns, it is just limited to the selected lines. We call this uniform sparse representation. To this end, our work in [19] performs line selection by applying trimming regularization techniques to eliminate one line at a time until a stable support is reached. After the DL process is completed, we perform anomaly detection: we test to see if a (new) data-item is anomalous by looking at its support and testing if it is included in the lines selected during training. If it is not, then it is labeled as an outlier. In our experiments we found that the set inclusion based anomaly detection technique can sometimes be a bit too conservative which encourages false positives.

One Class Support Vector Machines (OC-SVM) [4] is the unsupervised variant of the standard SVM problem where the inliers are separated from the outliers by the hyperplane separating the origin from most data points. OC-SVM is a standard anomaly detection algorithm that has been extended and integrated with multiple other methods including autoencoder neural networks [20, 21].

*Contributions.* The main contributions in our paper are listed as:

(*i*) We propose a new anomaly detection model that adapts and extends the DL formulation to include the OC-SVM objective, where OC-SVM takes as input the DL-based sparse representations. This will produce a strong coupling between sparse representations and vector machines which will permit us to replace testing for outliers via set inclusion by a simple OC-SVM call. The combination of these two methods will enable us to merge sparse feature extraction and anomaly

detection into a single model that will improve the generalization of the detection process.

(*ii*) Besides introducing a new anomaly detection model, we also provide theoretical results and algorithms for our two problem formulations.

(*iii*) We further extend both objectives to more general setting that allows the use of kernel functions.

(*iv*) We provide extensive numerical tests that show that the empirical quality of our schemes exceeds the one of other shallow widely known learning algorithms.

(*v*) Finally, we complete the comparative numerical tests with deep autoencoders showing empirically that our methods provide similar or better results.

*Outline.* In Section 2 we discuss and connect our proposal to the state of the art and continue to describe the anomaly detection methodology and general approach in Section 3. Section 4 presents our main theorem describing the iterations of the mixed DL-OCSVM objective. In Section 5 we introduce non-linearity via kernel methods and adapt the algorithms and the theoretical framework to the new formulations. To validate our theory, in Section 6 we present extensive numerical simulations against the state of the art comprising of tests on real-data from multiple domains. We conclude in Section 7.

*Notations.* Let $a_i$ be the $i$-th column, $a^j$ the $j$-th line and $a_{ij}$ an element of matrix $A$. Where no indices are present, and unless otherwise specified, $a$ represents a column vector and $a^T$ a line vector. As a general rule we use lowercase for vectors and uppercase for matrices. We denote with $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$ the Frobenius norm and with $\|a\|$ the $\ell_2$ vector norm. The pseudo-norm $\|\cdot\|_0$ counts the number of nonzero elements of a vector. We denote the restriction to column normalized matrices as $\mathcal{N}_{m,n} = \{D \in \mathbf{R}^{m\times n} : \|d_j\| = 1, \forall j \in [n]\}$, where $[n] = \{1, \cdots, n\}$ for $n \geq 1$.

## 2. Preliminaries

We are interested in the dictionary learning problem where the training data set $Y \in \mathbf{R}^{m\times N}$ has full-rank and the dictionary matrices have normed columns (also called

atoms). The standard dictionary learning problem is

$$D^\star, X^\star = \arg\min_{X, D \in \mathcal{N}_{m,n}} \|Y - DX\|_F^2 \tag{1}$$

where $d_j$ is the $j$-th atom of dictionary $D \in \mathbf{R}^{m \times n}$ and $X \in \mathbf{R}^{n \times N}$ is the sparse representation matrix with support $\Omega$ such that $X_\Omega = \{x_{ij} \mid x_{ij} \neq 0 : \forall i \in [n], j \in [N]\}$. Our current work assumes that the support $\Omega$ is either given or obtained through a good sparse representation algorithm such as Orthogonal Matching Pursuit (OMP) [22].

Problem (1) is hard and current approaches are usually based on the K-SVD algorithm [23] which is a block-coordinate descent algorithm iterating over pairs of atoms $d_i$ and their corresponding rows $x^i \in X$ solving

$$(d_i^+, (x^i)^+) = \arg\min_{d_i, x^i : \|d_i\| = 1} \frac{1}{2} \left\| d_i x^i - R \right\|_F^2 \tag{2}$$

where $R = \left[ Y - \sum_{j \neq i} d_j x^j \right]_{\Omega^i}$ is the error $E = Y - DX$ with the $i$-th pair removed and restricted only to the signals in $Y$ that use atom $d_i$ in their representation. This implies that $x^i$ in (2) contains only the non-zero entries of row $i$ from $X$ which corresponds to the support from row $i$ of $\Omega$. Indeed, this changes nothing in our computations as including the zero entries would only generate zero columns in $d_i x^i$ which can not reduce the residual $R$. Thus, K-SVD iterations update the atoms and their associated coefficients in $X$ without altering $\Omega$. We will use this fact and the restricted form throughout the paper.

We are also interested in the One-Class Support Vector Machine (OC-SVM) algorithm [4] which is the unsupervised anomaly detection variant of the well known SVM algorithm that has recently shown good results when coupled with other machine learning techniques [21, 20, 24]. The primal OC-SVM objective is

$$\min_{\omega, \rho, \xi} \quad \|\omega\|^2 - \rho + \frac{1}{CN} \sum_{i=1}^{N} \xi_i \quad \text{s.t.} \quad \omega^T x_i \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in [N] \tag{3}$$

where $(\omega, \rho)$ are the separating hyperplane learned by OC-SVM from training data $X$, $\xi$ are the slack variables and $C$ is a regularization parameter. In our work we will use Lagrange multipliers to solve the OC-SVM problem. In Section 4 we show that this also helps us fuse the DL and OC-SVM objectives as described around (12).

Our paper simultaneous performs unsupervised dictionary learning on the given data and trains OC-SVM on the resulting sparse representation by combining the objective of the two algorithms as described in Section 3. The goal is to aid the OC-SVM separation process by first projecting the data in the sparse dictionary subspaces where anomalies are spread further apart from normal data points than in the original space; both the dictionary and the support vector machine are trained simultaneously for this task.

### *2.1. Uniform support for unsupervised anomaly detection*

When obtaining $\Omega$, sparse representation algorithms in general, and OMP in particular, focus on column based sparsity (e.g. each $x_i \in X$ is at most $s$-sparse). Of important note here is the fact that OMP provides unstructured sparsity: given $x_i$ and $x_j$ from $X$, it is with high probability that the two lie in different subspaces and thus their support differs; on the other hand a row from $X$ has any number of zeros and non-zeros and differs completely from the next. In the rare case where a row of $X$ is zero, this implies that the corresponding atom from $D$ is unused, thus some K-SVD variants perform atom replacement after each iteration (2) if $(x^i)^+ = 0$ in the hope that the new atom will be useful at the next iteration when recomputing $\Omega$.

This is unlike structured sparisty approaches [25] such as Principal Component Analysis (PCA) [26] or Joint Sparse Representation (JSR) [27] where each $x \in X$ lies on the same subspace. Here the rows of $X$ are either zero or filled with non-zeros.

In our pursuit for unsupervised anomaly detection, we introduced a mix of the two approaches in [19], that attacks a regularized formulation of the (1) objective

$$D^\star, X^\star = \arg\min_{D,X} \|Y - DX\|_F^2 + \beta \sum_i \phi(\left\|x^i\right\|) \tag{4}$$

where $\phi$ is a sparse regularizer on the rows in $X$. Please note the following particular forms of $\phi$: when $\phi(z) = z$ we recover the widely known sparse penalty $\|X\|_{2,1}$ also known as the $\ell_{2,1}$-norm; similarly $\phi(z) = \|z\|_0$ leads to the $\ell_{2,0}$-norm and $\phi(z) = \ell_\epsilon(z) := \min\{|z|, \epsilon\}$ is the truncated $\ell_{2,\varepsilon}$-norm. The resulting algorithm starts from an unstructured support obtained through OMP and then performs KSVD-like iterations

that follow

$$(d_i^+, (x^i)^+) = \arg \min_{d_i : \|d_i\|=1, x^i} \frac{1}{2} \left\| d_i x^i - R^k \right\|_F^2 + \beta \phi(\left\| x^i \right\|_2) \tag{5}$$

When $\phi(z) = z$, if $\sigma_1 \geq \beta$ the iteration has the closed form $(d_i, x^i) = \left(u_1, (\sigma_1 - \beta) v^1\right)$ otherwise $(d_i, x^i) = (d_i, 0)$ where $\sigma_1 u_1 v^1$ is the first SVD term of $R$. When the condition is not met, the entire row is zeroed in PCA fashion resulting in a uniform support. Please note that at the end the remaining non-zero rows, denoted with $\mathcal{I}$, still have an unstructured sparsity pattern (i.e. $X_{\mathcal{I}}$ resembles an OMP generated sparsity pattern). This process is called uniform sparse representation and, as described in the Introduction, given a test point $\tilde{y}$ whose sparse representation $\tilde{x}$ is obtained through the learned dictionary $D$ via OMP, we perform anomaly detection through testing if the support of $\tilde{x}$ is included in $\mathcal{I}$.

In this paper we integrate the OC-SVM objective with the uniform support approach in (4) in order to obtain an unsupervised self-contained anomaly detection method that improves and provides a more theoretically sound anomaly detection technique. Also, due to the minor numerical differences between sparsity regularizers $\phi$ from [19], our theorems will only follow the $\ell_{2,1}$ penalty where $\phi(z) = z$; the non-convex $\ell_{2,0}$ and $\ell_{2,\varepsilon}$ regularizers are thus omitted but can be easily adapted in our proposed framework from the following sections.

### *2.2. Supervised pair learning with vector machines*

To our knowledge, there are few works that attempted to integrate dictionary learning with support vector machines. In [6], unlike us, the authors tackle the general supervised multi-class machine learning setting. The dictionary learning part is based on dictionary pair learning (DPL) for classification [28]

$$D^\star, P^\star = \arg\min_{D,P} \sum_{k=1}^{K} \|Y_k - D_k P_k Y_k\|_F^2 + \gamma \left\| P_k \bar{Y}_k \right\|_F^2 \tag{6}$$

where we are given $K$ classes with a labeled dataset $Y$ such that the data $Y_k$ belong to class $k$ and the $\bar{Y}_k$ data do not. For each class we learn two dictionaries. $D_k \in \mathbf{R}^{m \times n_k}$ is the synthesis dictionary from (1) that brings column sparsity in class $k$ representations $X_k \in \mathbf{R}^{n_k \times N}$. In the DPL formulation, the representations $X$ are no longer obtained

through a separate sparse representation algorithm, but instead through another dictionary learning problem such that $X_k = P_k Y_k$. The $P_k \in \mathbf{R}^{n_k \times m}$ dictionary is the analysis dictionary with most of its rows orthogonal to any given signal from $Y_k$.

In (6), the resulting representations $X$ follow a diagonal block form, $\|Y_k - D_k P_k Y_k\|_F^2 \leq \|Y_i - D_k P_k Y_i\|_F^2 \quad \forall i \neq k$, which can be viewed as clustering through regularization. Given a test point $z$, classification is performed through $c = \arg\min_k \|z - D_k P_k z\|^2$.

The authors in [6] extend the DPL objective to combine support vector machines with dictionary learning for general supervised classification tasks

$$D^\star, P^\star, X^\star, \omega^\star, \rho^\star = \arg\min_{D,P,X,\omega,\rho} \sum_{k=1}^{K} \|Y_k - D_k X_k\|_F^2 + \gamma_1 \|P_k Y_k - X_k\|_F^2 + \gamma_2 \left\|P_k \bar{Y}_k\right\|_F^2 + \gamma_3 g(X_k, h^k, \omega_k, \rho_k) + \gamma_4 \|P_k\|_F^2 . \quad (7)$$

Here we are presented with a minor adaptation of the DPL problem that is split among the first three terms such that the first term is the standard DL problem (1) providing $X_k$ to the second term that learns the analysis dictionary based on the $X_k = P_k Y_k$ DPL factorization. In the fourth term $g$ is the standard SVM function applied on the representations $X$ where $h$ are the labels and $(\omega, \rho)$ the separating hyperplane. The last term is a penalty on the magnitude of the coefficients from $P$. To solve (7), the authors employ an alternating minimization (AM) scheme on each of the variables. Given a test point $z$, classification is performed through $c = \arg\min_k \alpha \|z - D_k P_k z\|^2 - \left(\omega_k^T P_k z + \rho_k\right)$ where $\alpha > 0$ is an extra hyper-parameter meant to balance the two terms.

### *2.3. Related work*

Although mostly under the supervised setting and with few works addressing the fusion of DL and SVM, there exists a lot of recent work on this topic which shows that there is an active community with a keen interest in the area.

We start off with existing work employing supervised DL with SVM. In [29] the authors revisit (7) and propose an ADMM-based approach for attaining the objective, while [30] imposes smoothness and label-based profiling on the sparse coefficients by augmenting (7). [31] is very similar to the work proposed in [6] but in the standard DL setting (1) with Fisher-based inter and intra-class discrimination [32]. [5] follows

the same approach but without Fisher discrimination. Although the SVM model is not taken into consideration when updating the representations, [33] proposes a convolutional Fisher DL supervised discrimination scheme that uses an SVM classifier at the end on the resulting representations.

Within the DPL family we first mention supervised classification works such as [34] that performs anomaly detection using DPL for process monitoring, [35] employing multi-class DPL in the latent space of an autoencoder architecture, [36] imposing structure on the multi-class DPL based algorithm. Lastly, tackling only the DPL without classification but with a direct impact on the above we mention [37] that uses FISTA for sparse representations and alternating schemes for learning the dictionaries, and also [38] which extends the supervised discriminative setting to the DPL formulation. Other non-linear convolutional or autoencoder DL-based supervised classification techniques can be found in [39, 40].

In the anomaly detection realm we found little existing work. [41] provides supervised learning where the training set contains labeled outliers and the standard DL problem is penalized with $\|O\|_{2,1}$ where $O$ is the set of outliers; the paper focuses solely on representation error and does not provide results obtained for outlier detection. [42] extends the former with an extra graph smoothing constraint; the paper does not report results on a well known outlier detection benchmark. In [43], the authors change the matrix norm in the standard DL problem in order to minimize the maximal error, a simple method that resembles the weak trimming approach; the paper does not provide a scheme for anomaly detection nor numerical results related to the AD task. A two-step method that learns a dictionary for normal time series and then uses the reconstruction errors as replacements for the slack variables from the OC-SVM objective is presented in [44]. Another method that tries to detect anomalies in telemetry data is introduced by He et al. [45], which integrate OC-SVM and DL and obtain a formulation that is similar to ours. They solve the problem in a different way, optimizing the variables individually, and do not address the DPL formulation as we do.

For brevity, we include here a wider selection of papers along with their contributions and limitations. The methods in these papers are not from the same class or field of work as our methods, but tackle similar tasks. The work in [46] integrates recon-

struction loss and Deep SVDD loss and computes anomaly scores using the distance of the points from the latent space to their mean center; the approach does not offer significant performance gains for complex datasets. Another SVDD approach is proposed in [47] where multiple SVDD models learn to discover the region where representations of normal samples are concentrated and combined with Adaboost in order to obtain a strong classifier; we note here the significant increase in training time. In [48] background and anomaly dictionaries are built using a method based on GMM and LOF and are further used in the context of the low rank and sparse representation model; the aproach also leads to large running time. Two autoencoders are used in [49] to reconstruct backgrounds and anomalies from hyperspectral images, integrating different attention mechanisms. Background and anomaly dictionaries are used for hyperspectral data decomposition. Here the larger number of target pixels sometimes leads to missed detections.

In this paper we are interested in the unsupervised anomaly detection topic for which we propose a novel method fusing DL and OC-SVM. In the following sections we will adapt and analyze both the standard DL and DPL formulations for this task together with their non-linear adaptation through kernel methods.

## 3. Anomaly Detection Methodology and Algorithms

Our general model minimizes the combined loss function

$$\mathcal{L}(Y, D, X, \omega, \rho, \xi) = F(Y, D, X) + G(X, \omega, \rho, \xi), \tag{8}$$

where $F$ represents the dictionary learning objective and, respectively, $G$ the one-class support vector machine (OC-SVM) objective. We can now define the base model of our current study. Let $F$ follow the anomaly detection with uniform sparse representation from (4) and let $G$ be the standard OC-SVM problem

$$F(Y, D, X) = \frac{1}{2} \|Y - DX\|_F^2 + \beta \sum_{i=1}^{n} \phi\left(\left\|x^i\right\|_2\right) \tag{9}$$

$$G(X, \omega, \rho, \xi) = \max_{\lambda \geq 0} \ \|\omega\|^2 - \rho + \frac{1}{CN} \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \lambda_i (\omega^T x_i - \rho + \xi_i). \tag{10}$$

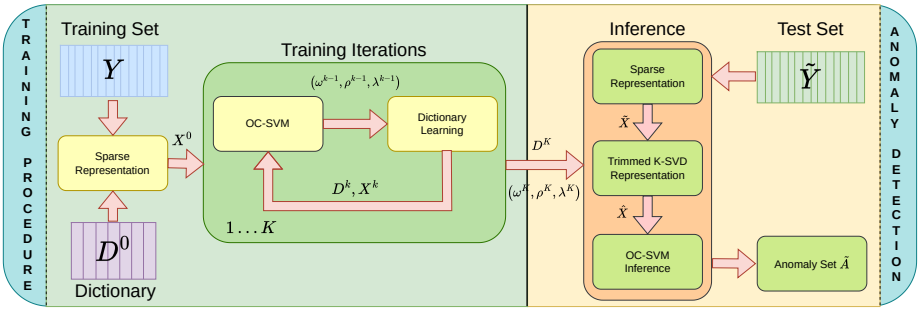


Figure 1: DL-OCSVM Anomaly Detection Diagram. In the left panel the training procedure starts with an initial dictionary $D^0$ and training set $Y$ and then proceeds to obtain the sparse representations $X$ on which the first OC-SVM model is trained. The tranining process iterates $K$ times between the DL routine, that uses the OC-SVM model variables to update the dictionary, and the OC-SVM routine, that uses the sparse representations from the DL model. In the right panel the DL and OC-SVM models are trained and inference is performed on the new set $\tilde{Y}$ by first obtaining the sparse representations with the trained dictionary $D^K$, trimming the representation via the dictionary update routine and running them through OC-SVM to obtain the anomaly set $\tilde{A}$.

Here we use the Lagrangian form of the OC-SVM problem where the representations $X$ produced through the dictionary learning process are used as support vectors, $C$ is the regularization parameter that represents the upper bound of the fraction of outliers (and the lower bound of the fraction of support vectors), $\omega$ and $\rho$ define the separating hyperplane, $\xi$ are slack variables and $\lambda$ the Lagrange multipliers. In the DPL formulation the objective will be modified accordingly as $X = PY$ implies that the variable $X$ is replaced by $P$ and the loss function becomes $\mathcal{L}(Y, D, P, \omega, \rho, \xi) = F(Y, D, P) + G(PY, \omega, \rho, \xi)$.

We propose an alternating minimization (AM) approach to solve the objective from (8). The steps necessary for our DL-OCSVM general anomaly detection scheme are organized in Algorithm 1. Steps 1–6 describe the training procedure. Step 2 computes the initial representations using the initial dictionary and column sparsity $s$. The resulting support $\Omega$ will be fixed during the rest of the procedure unless an entire row from $X$ is zeroed. Step 3 trains an initial OC-SVM model based on the initial representations. The AM iterations from steps 5 and 6, also called outer iterations, are performed $K$-times by step 4. Step 5 performs the inner-iterations of the DL process mixed with the OC-SVM objective and afterwards, in step 6, the OC-SVM model is updated with

**Algorithm 1:** DL-OCSVM: Anomaly Detection Scheme

**Data:** train set $Y$, test set $\tilde{Y}$, dictionary $D^0$, sparsity $s$, iterations K

**Result:** final dictionary $D$ and OC-SVM model $(\omega, \rho, \xi, \lambda)$

1 *Training Procedure*

2 Representation: apply OMP to obtain $X^0$ and its support $\Omega$

3 Support vectors: model $(\omega^0, \rho^0, \xi^0, \lambda^0)$ based on $X^0$

4 **for** $k \in \{1, \ldots, K\}$ **do**

5 DL: inner routine that learns $D^k$ and $X^k$ with fixed $\omega^{k-1}, \lambda^{k-1}$

6 Support vectors: model $(\omega^k, \rho^k, \xi^k, \lambda^k)$ based on fixed $X^k$

7 Inference: build anomalies set $\mathcal{A}$ based on OC-SVM at iteration $K$

8 *Anomaly Detection*

9 Representation: inner routine that obtains $\tilde{X}$ from $\tilde{Y}$

10 Inference: build anomalies set $\tilde{\mathcal{A}}$ using OC-SVM trained model on $\tilde{X}$

the new representations from the previous step.

There are two approaches to unsupervised anomaly detection. The first approach trains a model on the entire dataset (steps 1–6) and then provides the found anomalies in set $\mathcal{A}$ (step 7). The second approach trains a model on a data subset and then in steps 8–10 checks the rest of the data for anomalies by performing sparse representation with the trained dictionary (step 9) and inferring on the resulting representation with the trained OC-SVM model (step 10). We will investigate both approaches, while the former may provide better results because it sees all the data in training, the later can provide faster training times due to the reduced data size and can be less prone to over-fitting errors. Given a test point $z$ from the dataset, OC-SVM inference (at step 7 or 10) computes $h = \text{sign}\left(\omega^T z - \rho\right)$. If $h \leq 0$ then we found an anomaly and we add it to set $\mathcal{A}$.

In Figure 1 we provide a diagram of the DL-OCSVM anomaly detection scheme connecting the steps from Algorithm 1 with the input and output data structures. The following sections will analyze the algorithm properties together with the DPL and

kernel extensions.

## 4. Uniform representations through regularization and OC-SVM

In this section we will study the adaptation of standard DL and the DPL formulations for uniform representations mixed with OC-SVM and their effect on the general loss $\mathcal{L}$.

### *4.1. Standard DL with $\ell_{2,1}$ regularization*

Given the mixed objective (8) with $\phi(z) = z$ in (9) we obtain

$$\mathcal{L}(Y, D, X, \omega, \rho, \xi) = \frac{1}{2}\|Y - DX\|_F^2 + \beta \sum_{i=1}^{n} \left\|x^i\right\|_2 + G(X, \omega, \rho, \xi), \tag{11}$$

where $x^i$ represents a line from $X$. Let us consider the original K-SVD iteration where each atom $d_i$ and line $x^i$, are updated in a cyclic manner, at once. Our mixed objective becomes:

$$(d_i^+, (x^i)^+) = \underset{d_i:\|d_i\|=1, x^i}{\arg\min} \frac{1}{2}\left\|d_i x^i - R\right\|_F^2 + \beta\left\|x^i\right\|_2 - \omega_i x^i \lambda, \tag{12}$$

where $R = Y - \sum_{j\neq i} d_j x^j$ is the approximation residual without the contribution of $d_i$ and $x^i$, and, from the OC-SVM objective, $\omega_i \in \omega$ is the element corresponding to $d_i$ and $\lambda$ the Lagrange multipliers. The success of K-SVD updates for sparse representation problems [23] relied upon the explicit form of the new iterates pair $(d^+, x^+)$ and the fast convergence to a good stationary point. Although the K-SVD extension developed in (12) does not share the former benefit that allows its easy computation, we further formulate the new iterates as the solution of tractable quadratic minimization problems.

**Theorem 1.** *Let $\phi(z) = z$ and $\nu$ the element of $\omega$ corresponding to the current atom $d_i$. Then the closed form solution of the K-SVD iteration for* (12) *is given by: let* $d^* = \underset{\|d\|=1}{\arg\min} \ -\frac{1}{2}\left\|R^T d + \nu\lambda\right\|_2^2$
*(i) If $\left\|R^T d^* + \nu\lambda\right\| \geq \beta$ then*

$$d_i^+ = d^* \tag{13}$$

$$(x^i)^+ = \left(1 - \frac{\beta}{\left\|R^T d_i^+ + \nu\lambda\right\|}\right)(R^T d_i^+ + \nu\lambda)^T, \tag{14}$$

(*ii*) *Otherwise, when* $\left\|R^T d^* + \nu\lambda\right\| < \beta$, $(d_i^+, (x^i)^+) = (d_i, 0)$.

*Proof.* For simplicity we drop the coordinate indices. The modified K-SVD iteration (12) becomes:

$$\min_{\|d\|=1} \min_x \frac{1}{2}\|x\|^2 - xR^T d + \frac{1}{2}\|R\|_F^2 + \beta\|x\| - \nu x\lambda,$$

Denote $x^*(d)$ the solution in $x$ for a fixed $d$. The first order optimality of the inner problem is given by: $0 \in x^*(d) - (R^T d + \nu\lambda) + \beta\partial\|\cdot\|(x^*(d))$. If $\left\|R^T d + \nu\lambda\right\| \leq \beta$ then $x^*(d) = 0$, otherwise the above relation reduces to:

$$\left(1 + \frac{\beta}{\|x^*(d)\|}\right) x^*(d)^T = R^T d + \nu\lambda. \tag{15}$$

Then, a first consequence of (15) is $\|x^*(d)\| = \left\|R^T d + \nu\lambda\right\| - \beta$. By using this identity again in (15) yields as a second consequence $\left(1 + \frac{\beta}{\left\|R^T d+\nu\lambda\right\|-\beta}\right) x^*(d)^T = R^T d + \nu\lambda$ which produces in this case the explicit form of the solution $x^*(d)$

$$x^*(d) = \left(1 - \frac{\beta}{\left\|R^T d + \nu\lambda\right\|}\right)(R^T d + \nu\lambda)^T. \tag{16}$$

Replacing both null and explicit form in the original minimization problem we get:

$$\begin{aligned}
&\frac{1}{2}\|x^*(d)\|^2 - x^*(d)R^T d + \beta\|x^*(d)\| - \nu x^*(d)\lambda \\
&= \begin{cases} -\frac{1}{2}\left(\left\|R^T d + \nu\lambda\right\| - \beta\right)^2 & \text{if } \left\|R^T d + \nu\lambda\right\| > \beta \\ 0 & \text{otherwise.} \end{cases} = -\frac{1}{2}\left(\left\|R^T d + \nu\lambda\right\| - \beta\right)_+^2.
\end{aligned}$$

where $(\cdot)_+ := \max\{0, \cdot\}$. Since $(\cdot - \beta)_+^2$ is nondecreasing, then the K-SVD subproblem reduces to computing $d^* = \arg\max_{\|d\|=1} \frac{1}{2}\left\|R^T d + \nu\lambda\right\|_2^2$, followed by computing $x^*(d^*)$ through (16).

□

**Remark 2.** *Computing* $(d^i)^+$ *with* (13) *reduces to maximizing a convex quadratic cost over the sphere, which results in a nonconvex constrained optimization problem commonly named as trust region subproblem [50]. Despite its nonconvex nature, there are various references stating that the global solution is computable through simple iterative schemes [51, 52, 50, 53, 54]. Given the singular value decomposition* $R = U\Sigma V^T$,

**Algorithm 2:** DL-OCSVM: DL inner routine (step 5 from Algorithm 1)

**Data:** train set $Y \in \mathbf{R}^{m \times N}$, $D^{k-1} \in \mathbf{R}^{m \times n}$, inner iteration k

**Result:** dictionary $D^k$ and representations $X^k$

1 Error: $E^k = Y - D^{k-1}X^{k-1}$

2 **for** $i \in \{1, \ldots, n\}$ **do**

3 Atom error: $R^k = E^k + d_i^{k-1} x^{i,k-1}$

4 Update: new $(d_i^k, x^{i,k})$ according to Theorem 1

5 New error: $E^k = R^k - d_i^k x^{i,k}$

*the subproblem from* (13)

$$\min_{\|d\|=1} -\frac{1}{2} \left\| R^T d + \nu\lambda \right\|_2^2 = -\frac{1}{2} d^T R R^T d - \nu d^T R\lambda + c,$$

*can be reformulated through a change of variables into* $\bar{d} := U^T d$ *to get:*

$$\min_{\|\bar{d}\|=1} -\frac{1}{2} \bar{d}^T \Sigma^2 \bar{d} - \bar{d}^T q + c,$$

*where* $q := \nu\Sigma V^T \lambda$. *As stated in [50, Corrolary 8], by solving the bi-dual convex problem:*

$$\min_{\sum_i y_i = 1} -\frac{1}{2} \sum_i \sigma_i^2 y_i - |q_i| \sqrt{y_i}. \tag{17}$$

*we recover the optimal* $\bar{d}_i^* = sign(q_i)\sqrt{y_i^*}$ *and further more* $d^* = U\bar{d}^*$. *The final problem* (17) *is convex and it has a single linear equality constraint.*

The exact computation of $d^+$ based on the reasoning from above is computationally dominated by the SVD procedure at cost $O(m^2 N)$. The processed program (17) has been particularly solved in [50] by a simple interior-point scheme in $O(m \ln(2m/\epsilon))$, where $\epsilon$ is the accuracy of the solution. For this reason, we used in practice a particular Power Method scheme that approximates the solution (13) and avoids the SVD step.

We are now ready to describe in Algorithm 2 the inner-iterations of the DL process (see step 5 in Algorithm 1). Given the initial dictionary and representations from the last outer iteration, step 1 initializes the overall representation error $E$. Step 2 starts the

inner iterations walking through all the dictionary atoms and their associated rows in $X$. In step 3 we compute the residual of the current pair and proceed to update the atom and its representations at step 4 following Theorem 1. Finally, in step 5, we update the error with the new atom and representations.

**Proposition 3.** *Let $\{D^k, X^k, \omega^k, \rho^k, \xi^k\}_{k\geq 0}$ be the sequence generated by the Algorithm 1 using the inner iterations of Algorithm 2. Then the following decrease holds*

$$\mathcal{L}(Y, D^{k+1}, X^{k+1}, \omega^{k+1}, \rho^{k+1}, \xi^{k+1}) \leq \mathcal{L}(Y, D^k, X^k, \omega^k, \rho^k, \xi^k) \quad \forall k \geq 0.$$

*Proof.*

$$\begin{aligned}
&\mathcal{L}(Y, D^k, X^k, \omega^k, \rho^k, \xi^k) = F(Y, D^k, X^k) + G(X^k, \omega^k, \rho^k, \xi^k) \\
&= \frac{1}{2}\left\|R^k - d_i^k x^{i,k}\right\|_F^2 + \beta\left\|x^{i,k}\right\| + \beta\sum_{j\neq i}\left\|x^{j,k}\right\| \\
&\quad + \left\|\omega^k\right\|^2 - \rho^k + \frac{1}{CN}\sum_i \xi_i^k - \omega_i^k x^{i,k}\lambda - \sum_{j\neq i}\omega_j^k x^{j,k}\lambda + \sum_j \lambda_j\left[-\rho^k + \xi_j^k\right] \\
&\overset{Theorem1}{\geq} \frac{1}{2}\left\|R^k - d_i^{k+1} x^{i,k+1}\right\|_F^2 + \beta\left\|x^{i,k+1}\right\| - \omega_i^k x^{i,k+1}\lambda + \beta\sum_{j\neq i}\left\|x^{j,k}\right\| \\
&\quad + \left\|\omega^k\right\|^2 - \rho^k + \frac{1}{CN}\sum_i \xi_i^k - \sum_{j\neq i}\omega_j^k x^{j,k}\lambda + \sum_j \lambda_j\left[-\rho^k + \xi_j^k\right] \\
&\overset{Theorem1}{\geq} \frac{1}{2}\left\|R^k - d_i^{k+1} x^{i,k+1} - d_{i+1}^{k+1} x^{i+1,k+1}\right\|_F^2 + \beta\left(\left\|x^{i,k+1}\right\| + \left\|x^{i+1,k+1}\right\|\right) \\
&\quad - \left(\omega_i^k x^{i,k+1} + \omega_{i+1}^k x^{i+1,k+1}\right)\lambda + \beta\sum_{j\neq i,i+1}\left\|x^{j,k}\right\| \\
&\quad + \left\|\omega^k\right\|^2 - \rho^k + \frac{1}{CN}\sum_i \xi_i^k - \sum_{j\neq i,i+1}\omega_j^k x^{j,k}\lambda + \sum_j \lambda_j\left[-\rho^k + \xi_j^k\right] \\
&\geq F(Y, D^{k+1}, X^{k+1}) + \left\|\omega^k\right\|^2 - \rho^k + \frac{1}{CN}\sum_i \xi_i^k - \sum_j \omega_j^k x^{j,k+1}\lambda \\
&\quad + \sum_j \lambda_j\left[-\rho^k + \xi_j^k\right] = F(Y, D^{k+1}, X^{k+1}) + G(X^{k+1}, \omega^k, \rho^k, \xi^k) \\
&\overset{\text{OC-SVM}}{\geq} F(Y, D^{k+1}, X^{k+1}) + G(X^{k+1}, \omega^{k+1}, \rho^{k+1}, \xi^{k+1}) \\
&= \mathcal{L}(Y, D^{k+1}, X^{k+1}, \omega^{k+1}, \rho^{k+1}, \xi^{k+1})
\end{aligned}$$

where $G(X^{k+1}, \omega^k, \rho^k, \xi^k)$ implied just an inference on $X^{k+1}$ with the existing model $(\omega^k, \rho^k, \xi^k)$ and $G(X^{k+1}, \omega^{k+1}, \rho^{k+1}, \xi^{k+1})$ implies an update of the support vectors with

**Algorithm 3:** DL-OCSVM: Anomaly Detection (steps 8–10 in Algorithm 1)

**Data:** test set $\tilde{Y} \in \mathbf{R}^{m \times \tilde{N}}$, dictionary $D \in \mathbf{R}^{m \times n}$, sparsity $s$,
and OC-SVM model $(\omega, \rho, \lambda)$

**Result:** anomalies $\tilde{\mathcal{A}}$

1 Representation: $\tilde{X} = \text{OMP}(\tilde{Y}, D, s)$

2 Error: $E = \tilde{Y} - D\tilde{X}$

3 **for** $i \in \{1, \ldots, n\}$ **do**

4 Atom error: $R = E + d_i \tilde{x}^i$

5 Trimming: zero $\tilde{x}^i$ if condition from (5) holds

6 **if** $sign\left(\omega^T \tilde{x}_i - \rho\right) \leq 0$ **then** $\tilde{\mathcal{A}} = \tilde{\mathcal{A}} \cup \{i\} \quad \forall i \in \tilde{N}$

the new signals $X^{k+1}$ and an inference afterwards thus making the last inequality true.

□

Proposition 3 and its proof can be easily adapted to the algorithms regarding the DPL formulation and the kernel variants from the following sections and so we will not revisit the topic.

Assume that the Training Procedure part from from Algorithm 1 steps 1–6 was already performed with the inner iterations of Algorithm 2 thus obtaining a trained DL-OCSVM model. We now detail the Anomaly Detection part from Algorithm 1 steps 8–10. First note that after training is completed the OC-SVM term $\omega_i x^i \lambda$ from (12) vanishes as new test data does not affect nor depend on the training data and the associated $\lambda$ parameters. Thus (12) resumes to problem (5) described in Section 2.1. Taking this in consideration during testing, for uniform representation we check the condition $\sigma_1 < \beta$ and zero the row $x$ when it holds as described below (5).

For clarity we include in Algorithm 3 all the necessary steps for performing anomaly detection with the trained model on a different set $\tilde{Y}$. Steps 1 and 2 produce the representations with the provided dictionary and compute the representation error similar to the initialization during training. Step 3 starts the inner iterations walking the dictionary atoms without updating them but instead computing the residual (step 4) and

zeroing the associated representations row if the condition from (5) holds. Finally, in step 6 we apply OC-SVM inference on the resulting sparse representations $\tilde{X}$.

### *4.2. Adapting the DPL formulation*

As we have seen in Section 2, the DPL formulation is often used in the literature mostly due to its efficient sparse representation computation: it is much faster to compute the matrix multiplication $PY$ than to apply a greedy algorithm such as OMP.

Thus let us investigate next the DPL formulation by modifying $F$ in (9) to include the $DPY$ factorization

$$F(Y, D, P) = \frac{1}{2} \|Y - DPY\|_F^2 + \beta \sum_{i=1}^{n} \phi\left(\left\|p^i Y\right\|_2\right) + \sum_{i=1}^{n} \alpha_i \left\|p^i Y\right\|_1 \tag{18}$$

where $D$ is the synthesis dictionary and $P$ is the analysis dictionary. In this formulation the sparse representations $X$ are obtained through the simple matrix multiplication $PY$. We add the third penalization term $\sum_{i=1}^{n} \alpha_i \left\|p^i Y\right\|_1$ in order to induce zeros on the rows of matrix $X$. This is helpful because, unlike multi-class classification schemes such as [28] and [6] which rely on group sparsity using per-class matrices $D_k$ and $P_k$ as discussed in Section 2.2, in our approach we have a single $D$ and $P$ and we rely on the individual sparse representations for discrimination.

**Remark 4.** ***Choosing $\alpha$ for the $\ell_1$ penalty.*** *We have $Ns$ nonzeros in $X$, $s$ per each column, out of the total $Nm$ elements of $X$. Let $\Omega$ be the initial support of $X$ generated by OMP and $\Omega^i$ the support of row $x^i$ such that $\varsigma_i = \left\|x^i\right\|_0 = |\Omega^i|$. We can set $\alpha \in \mathbf{R}^n$ as the unit vector whose elements are built from the row-wise $\ell_0$-norm of $X$ as $\alpha_i = \varsigma_i / \sqrt{\sum_k \varsigma_k^2}$ which will then impose row-sparsity according to $\Omega$. To cope with the other parameters in* (18) *we can introduce a common parameter $\gamma$ and rewrite the last term as $\gamma \sum_{i=1}^{n} \alpha_i \|p_i Y\|_1$.*

In K-SVD fashion, we approach (18) through the update of each synthesis atom $d$ in $D$ and its corresponding analysis atom $p$ from $P$ which produces the sparse representation line $x = pY$ from $X$. For the convex $\ell_{2,1}$ regularization case where $\phi(z) = z$, the modified K-SVD iteration becomes

$$(d_i^+, (p^i)^+) = \underset{d_i : \|d_i\|=1, p^i}{\arg\min} \frac{1}{2} \left\|d_i p^i Y - R\right\|_F^2 + \beta \left\|p^i Y\right\|_2 + \alpha \left\|p^i Y\right\|_1 - \omega_i p^i Y \lambda, \tag{19}$$

where $R = Y - \sum_{j \neq i} d_j p^j Y$ is the approximation residual without the $(d_i, p^i)$ pair. The solution of (19) is obtained through the following theorem.

**Theorem 5.** *Assume $Y^\dagger = Y^T(YY^T)^{-1}$ exists and denote $H_Y = Y^\dagger Y$. Then the closed form solution of the K-SVD iteration for* (18) *is:*

$$(d^+, \tau_1(d^+), \tau_2(d^+)) = \min_{\|d\|=1} \max_{\substack{\|\tau_2\|_2 \leq 1, \\ \|\tau_1\|_\infty \leq 1}} -\frac{1}{2} \left\| R^T d + \nu\lambda - \alpha\tau_1 - \beta\tau_2 \right\|_{H_Y}^2 \tag{20}$$

$$p^+ = (R^T d^+ + \nu\lambda - \beta\tau_2(d^+) - \alpha\tau_1(d^+))^T Y^\dagger, \tag{21}$$

*Proof.* Let $p$ be an arbitrary line in matrix $P$. Therefore, substituting the representation line $x = pY$ in the K-SVD iteration we obtain:

$$\begin{aligned}
&\min_{\|d\|=1} \min_{x=pY} \frac{1}{2} \|x\|^2 - x\left(R^T d + \nu\lambda\right) + \beta \|x\| + \alpha \|x\|_1 && (22)\\
&= \min_{\|d\|=1} \min_{p} \frac{1}{2} \|pY\|^2 - pY\left(R^T d + \nu\lambda\right) + \beta \|pY\| + \alpha \|pY\|_1 \\
&= \min_{\|d\|=1} \min_{p} \frac{1}{2} \|pY\|^2 - pY\left(R^T d + \nu\lambda\right) + \max_{\|\tau_2\| \leq 1} \beta pY\tau_2 + \max_{\|\tau_1\|_\infty \leq 1} \alpha pY\tau_1 \\
&= \min_{\|d\|=1} \min_{p} \max_{\|\tau_2\| \leq 1, \|\tau\|_\infty \leq 1} \frac{1}{2} \|pY\|^2 - pY\left(R^T d + \nu\lambda\right) + \beta pY\tau_2 + \alpha pY\tau_1 \\
&= \min_{\|d\|=1} \max_{\|\tau_2\| \leq 1, \|\tau\|_\infty \leq 1} \min_{p} \frac{1}{2} \|pY\|^2 - pY\left(R^T d + \nu\lambda - \beta\tau_2 - \alpha\tau_1\right) && (23)\\
&= \min_{\|d\|=1} \max_{\|\tau_2\| \leq 1, \|\tau\|_\infty \leq 1} -\frac{1}{2} \left\| H_Y(R^T d + \nu\lambda - \beta\tau_2 - \alpha\tau_1) \right\|^2, \\
&= \min_{\|d\|=1} \max_{\|\tau_2\| \leq 1, \|\tau\|_\infty \leq 1} -\frac{1}{2} \left\| R^T d + \nu\lambda - \beta\tau_2 - \alpha\tau_1 \right\|_{H_Y}^2, && (24)
\end{aligned}$$

where in (23) we used the first order optimality conditions over variable $p$ and the form of the solution $p(d) = (R^T d + \nu\lambda - \beta\tau_2(d) - \alpha\tau_1(d))^T Y^\dagger$. □

Since the minimization step (19) is nonconvex and nonsmooth, we rely in our tests on the approximated updates (20)-(21). For approximation, we used a projected gradient method with constant stepsize.

Our algorithms require minor modifications to accommodate the DPL form. The $(d, x)$ update from Algorithm 2 step 4 is replaced with updating the $(d, p)$ pair as in Theorem 5. For anomaly detection with a trained model, we no longer require steps 2 and 4 in Algorithm 3 and during trimming in step 5 we zero the lines in $\tilde{X}$ where $\|p\tilde{Y}\|$

is (almost) null. In practice, following K-SVD, $R$ and $Y$ contain only the signals which use atom $d$ in their representation.

## 5. Kernel formulation

Kernel functions $\varphi : \mathbf{R}^m \to \mathbf{R}^{\tilde{m}}$ are used to induce non-linearity in the feature space of $Y \in \mathbf{R}^{m\times N}$ leading to the higher dimensional space $\varphi(Y) \in \mathbf{R}^{\tilde{m}\times N}$ where $\tilde{m} \gg m$. While this potentially improves our algorithms modeling capabilities, it also comes with significant computational costs that can be reduced through the *kernel trick* when Mercer functions are used. If a kernel takes us to possibly infinite $\tilde{m}$ dimensional feature spaces, the kernel trick helps us reduce computations to $O(N)$ instead. An improvement, but still far from our initial $m$ dimensionality.

### *5.1. Kernel Dictionary Learning*

In order to employ the kernel trick, the kernel dictionary learning problem rewrites the dictionary as $D = \tilde{D} + D_\perp$ with $\tilde{D} \in \text{span}(\varphi(Y))$ and $D_\perp$ its orthogonal complement such that $\varphi(Y)^T D_\perp = 0$ and $\tilde{D}^T D_\perp = 0$. As shown in [3, Ch.9] inserting this in the standard dictionary learning problem (1) leads to the equivalent problem

$$\min_{A,X} \|\varphi(Y)(I - AX)\|_F^2 \tag{25}$$

where $D_\perp = 0$ and the dictionary $\tilde{D}$ is rewritten as $D = \varphi(Y)A$ with $A \in \mathbf{R}^{N\times n}$. Indeed $\|Y - DX\|_F^2 = \|\varphi(Y)(I - AX)\|_F^2$, whereas in the linear case (i.e. $\varphi(Y) = Y$) it does not make sense to work with $AX$ as it involves more computation with no benefit, for the kernel formulation it actually reduces algorithmic complexity because when we unpack (25) we form $\mathcal{K} = \varphi(Y)^T\varphi(Y)$ with $\mathcal{K} \in \mathbf{R}^{N\times N}$ which enables the kernel trick. We denote with $k_{ij} = \varphi(y_i)^T\varphi(y_j) = \kappa(y_i, y_j)$ the elements of $\mathcal{K}$.

Let $z \in \mathbf{R}^m$ be a given test point, also let $A \in \mathbf{R}^{N\times n}$ be the trained kernel dictionary from (25) and let $\zeta = \kappa(y_i, z) = \{\zeta_i \mid \zeta_i = \varphi(y_i)^T\varphi(z),\ \forall i \in [N]\}$. Then the resulting representation error is $e = \varphi(z) - \varphi(Y)Ax = \varphi(z) - \varphi(Y)A_I x_I$ with $x_I = \{x_i \mid x_i \neq 0\}$. For OMP atom selection [55] we never compute $e$ and instead we use the equivalent quantity to evaluate the correlation $D^T e = D^T[\varphi(z) - \varphi(Y)Ax] =$

$A^T\varphi(Y)^T[\varphi(z) - \varphi(Y)Ax] = A^T(\zeta - \mathcal{K}Ax)$. For multiple test points $\tilde{Y} \in \mathbf{R}^{m\times\tilde{N}}$, where $\tilde{k_j} = \{\tilde{k}_{ij} \mid \tilde{k}_{ij} = \varphi(y_i)^T\varphi(\tilde{y}_j),\ \forall i \in [N]\}$ with $j \in [\tilde{N}]$, we arrive at $D^TE = A^T(\tilde{\mathcal{K}} - \mathcal{K}AX) = A^T\tilde{\mathcal{K}} - A^T\mathcal{K}AX$. This can be solved by using OMP$(A^T\tilde{\mathcal{K}}, A^T\mathcal{K}A, s)$ in step 1 of Algorithm 3 where $Y \leftarrow A^T\tilde{\mathcal{K}}$ and $D \leftarrow A^T\mathcal{K}A$ .

In Step 2, the representation error also needs adjusting. For the test set $\tilde{Y}$, the representation error becomes $E = \varphi(\tilde{Y}) - D\tilde{X} = \varphi(Y)\left(\Theta - A\tilde{X}\right)$ where $\Theta$ is the expansion coefficients of $\varphi(\tilde{Y})$ in the span of $\varphi(Y)$ such that $\Theta = \varphi(Y)^\dagger\varphi(\tilde{Y}) = \left(\varphi(Y)^T\varphi(Y)\right)^{-1}\varphi(Y)^T\varphi(\tilde{Y}) = \mathcal{K}^{-1}\tilde{\mathcal{K}}$. In kernel DL algorithms, the representation error is replaced with just $E = \Theta - A\tilde{X}$, working in the A space not in the feature space [3, Ch.9.4], which leads to $E = \mathcal{K}^{-1}\tilde{\mathcal{K}} - A\tilde{X}$. During training $\tilde{\mathcal{K}} \leftarrow \mathcal{K}$ and we modify the above accordingly; of special note is the training error $E = I - AX$ which circles back to the objective in (25).

The standard Kernel K-SVD iteration follows $\min_{a,x}\|\varphi(Y)(R - ax)\|_F^2$ such that $\left\|\mathcal{K}^{\frac{1}{2}}a\right\| = 1$. The normalization constraint follows from expanding the Frobenius norm in (25) $(A^T\varphi(Y)^T)(\varphi(Y)A) = A^T\mathcal{K}A = (A^T\mathcal{K}^{\frac{1}{2}})(\mathcal{K}^{\frac{1}{2}}A)$. The kernel dictionary learning objective $F$ becomes

$$F(Y, D, X) = \frac{1}{2}\|\varphi(Y)(I - AX)\|_F^2 + \beta\sum_i \phi\left(\left\|x^i\right\|_2\right) \tag{26}$$

Note that $G$ is not affected by this as the OC-SVM problem deals with the sparse representations in $X$.

For the $\ell_{2,1}$ regularization from Section 4.1 we take into consideration only the $(a, x)$ pair

$$\begin{aligned}
&\frac{1}{2}\|\varphi(Y)(R - ax)\|_F^2 + \beta\|x\|_2 = \frac{1}{2}Tr[(R - ax)^T\varphi(Y)^T\varphi(Y)(R - ax)] + \beta\|x\|_2 \\
&\quad = \frac{1}{2}Tr(R^T\mathcal{K}R - R^T\mathcal{K}ax - x^Ta^T\mathcal{K}R + x^Ta^T\mathcal{K}ax) + \beta\|x\|_2 \\
&\quad = \frac{1}{2}(Tr(R^T\mathcal{K}R) - 2xR^T\mathcal{K}a + a^T\mathcal{K}a\|x\|_2^2) + \beta\|x\|_2
\end{aligned}$$

which, together with the objective from $G$, leads to the modified K-SVD iteration

$$(a_i^+, (x^i)^+) = \underset{a_i:\left\|\mathcal{K}^{\frac{1}{2}}a_i\right\|=1, x^i}{\arg\min}\ \frac{1}{2}\left\|x^i\right\|^2 - x^i(R^T\mathcal{K}a_i + \nu\lambda) + \beta\left\|x^i\right\| \tag{27}$$

where we used the fact that $a_i^T\mathcal{K}a_i = 1$. We show how to solve this iteration as a Corollary to Theorem 1.

**Corollary 6.** *Let $\phi(z) = z$, $\nu$ the corresponding element of $\omega$ and $\mathcal{K}$ the kernel matrix. Then the closed form solution of the K-SVD iteration for* (27) *is: if* $\left\|R^T\mathcal{K}a + \nu\lambda\right\| \geq \beta$ *then*

$$a^* = \mathcal{K}^{-\frac{1}{2}} \arg\max_{\|f\|=1} \frac{1}{2}\left\|R^T\mathcal{K}^{\frac{1}{2}}f + \nu\lambda\right\|_2^2, \tag{28}$$

$$x^*(a^*) = \left(1 - \frac{\beta}{\left\|R^T\mathcal{K}a^* + \nu\lambda\right\|}\right)(R^T\mathcal{K}a^* + \nu\lambda)^T, \tag{29}$$

*otherwise* $(d, x) = (d, 0)$. *Where solving for $a^*$ reduces to a trust region problem.*

*Proof.* Denoting $x^*(a)$ the solution in $x$ for a fixed $a$, the first order optimality of the inner problem reduces implies:

$$0 \in x^*(a)^T - R^T\mathcal{K}a - \nu\lambda + \beta\partial\left\|x^*(a)\right\| \tag{30}$$

Notice that if $\left\|R^T\mathcal{K}a + \nu\lambda\right\| \leq \beta$ then $x^*(a) = 0$. The proof follows that of Theorem 1 leading to (29) and

$$a^* = \arg\max_{\left\|\mathcal{K}^{\frac{1}{2}}a\right\|=1} \frac{1}{2}\left\|R^T\mathcal{K}a + \nu\lambda\right\|_2^2 \tag{31}$$

which can be rewritten by using $f = \mathcal{K}^{\frac{1}{2}}a$ as:

$$f^* = \arg\max_{\|f\|=1} \frac{1}{2}\left\|R^T\mathcal{K}^{\frac{1}{2}}f + \nu\lambda\right\|_2^2 \tag{32}$$

and then we can recover $a^*$ from $\mathcal{K}^{\frac{1}{2}}a^* = f^*$.

Given the eigendecomposition of the PSD matrix $\mathcal{K} = U\Lambda U^T$ where $\Lambda$ is diagonal, then $\mathcal{K}^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^T$ and $a^* = U\Lambda^{-\frac{1}{2}}U^T f^*$

□

To conclude, we sum-up the adaptations required in our algorithms for the kernel DL formulation. As discussed, Algorithm 2 step 1 will use the error $E = I - AX$ which will also reflect in the other steps involving the residual $R$. Also, in step 4 the update will regard the $(a, x)$ pairs of Corollary 6. When performing anomaly detection with a trained model, we have to adapt the following steps of Algorithm 3: step 1 will call OMP($A^T\tilde{\mathcal{K}}, A^T\mathcal{K}A, s$), step 2 will compute the error as $E = \mathcal{K}^{-1}\tilde{\mathcal{K}} - A\tilde{X}$ and during trimming in step 5 we will zero the lines if the corresponding condition from Corollary 6 holds.

*5.2. Kernel DPL*

The kernel version of the DPL problem requires using kernels for the sparse representations as well. Let us factorize, without loss of generality, $P = BY^T$, in a fashion similar to the factorization $D = YA$ used in kernel DL. In this way each row $p^i$ of $P$ is expressed as a linear combination of the signals in $Y$, with the coefficients $b^i$ as the $i$-th row of the tall matrix $B$, $p^i = b^i Y^T$. The DPL formulation (18) becomes

$$F(Y, D, B) = \frac{1}{2} \left\| Y - DBY^T Y \right\|_F^2 + \beta \sum_{i=1}^{n} \phi \left( \left\| b^i Y^T Y \right\|_2 \right) + \sum_{i=1}^{n} \alpha_i \left\| b^i Y^T Y \right\|_1 . \tag{33}$$

In the kernel DL setting, using kernels $\varphi(Y)$, this becomes

$$F(Y, A, B) = \frac{1}{2} \left\| \varphi(Y)(I - AB\varphi(Y)^T \varphi(Y)) \right\|_F^2 + \tag{34}$$

$$+ \beta \sum_{i=1}^{n} \phi \left( \left\| b^i \varphi(Y)^T \varphi(Y) \right\|_2 \right) + \sum_{i=1}^{n} \alpha_i \left\| b^i \varphi(Y)^T \varphi(Y) \right\|_1 \tag{35}$$

$$= \frac{1}{2} \left\| \varphi(Y)(I - AB\mathcal{K}) \right\|_F^2 + \beta \sum_{i=1}^{n} \phi \left( \left\| b^i \mathcal{K} \right\|_2 \right) + \sum_{i=1}^{n} \alpha_i \left\| b^i \mathcal{K} \right\|_1 \tag{36}$$

Note that equation (36) is the kernel version of (18), with $PY$ replaced now by $B\mathcal{K}$.

When solving iteratively in K-SVD fashion, the subproblem resembles a combination of (22) and (27), with $x = b\mathcal{K}$:

$$(a_i^+, (x^i)^+) = \underset{a_i : \left\| \mathcal{K}^{\frac{1}{2}} a_i \right\| = 1, x^i : x^i = b^i \mathcal{K}}{\arg\min} \frac{1}{2} \left\| x^i \right\|^2 - x \left( R^T \mathcal{K} a_i + \nu\lambda \right) + \beta \left\| x^i \right\| + \alpha \left\| x^i \right\|_1 \tag{37}$$

Note that (37) is in fact identical to (22) with $d \leftarrow \mathcal{K}^{\frac{1}{2}} a$, $x \leftarrow b\mathcal{K}$, $R \leftarrow \mathcal{K}^{\frac{1}{2}} R$ and $Y \leftarrow \mathcal{K}$, and thus the solution can be found with the same procedure. For completeness, we formalize the result below.

**Corollary 7.** *Let $\mathcal{K}^\dagger$ denote the Moore-Penrose pseudoinverse of $\mathcal{K}$, and denote $H_\mathcal{K} = \mathcal{K}\mathcal{K}^\dagger = \mathcal{K}^\dagger\mathcal{K}$ (since $\mathcal{K}$ is symmetric). Then the solution of the inner K-SVD subproblem* (37) *for the kernel DPL global objective* (36) *is the following:*

$$(d^+, \tau_1(d^+), \tau_2(d^+)) = \min_{\|d\|=1} \max_{\substack{\|\tau_2\|_2 \le 1, \\ \|\tau_1\|_\infty \le 1}} -\frac{1}{2} \left\| R^T \mathcal{K}^{\frac{1}{2}} d + \nu\lambda - \alpha\tau_1 - \beta\tau_2 \right\|_{H_\mathcal{K}}^2 \tag{38}$$

$$a^+ = \mathcal{K}^{-\frac{1}{2}} d^+ \tag{39}$$

Table 1: DL-OCSVM important variables and hyper-parameter summary.

| Algorithms | Notation | Description |
|---|---|---|
| DL, DPL, KDL, KDPL | $\beta$ | uniform support regularization and threshold |
| DPL, KDPL | $\alpha$ | $\ell_1$ sparsity regularization on rows $p^i Y,\ i \in [n]$ |
| OCSVM | $C$ | regularization and anomalies upper-bound fraction |
| | $\xi$ | slack variables |
| | $\rho$ | separating hyperplane offset |
| DL, DPL, KDL, KDPL, OCSVM | $\omega$ | separating hyperplane normal vector |
| | $\lambda$ | Lagrange multipliers |

$$b^+ = (R^T \mathcal{K}^{\frac{1}{2}} d^+ + \nu\lambda - \beta\tau_2(d^+) - \alpha\tau_1(d^+))^T \mathcal{K}^\dagger \tag{40}$$

$$\tag{41}$$

*As a consequence, the optimal x is:*

$$x^+ = b^+ \mathcal{K} = (R^T \mathcal{K}^{\frac{1}{2}} d^+ + \nu\lambda - \beta\tau_2(d^+) - \alpha\tau_1(d^+))^T H_{\mathcal{K}} \tag{42}$$

*Proof.* The inner K-SVD sub-problem (37) is identical to (22) with the following change of variables: $d \leftarrow \mathcal{K}^{\frac{1}{2}} a$, $R \leftarrow \mathcal{K}^{\frac{1}{2}} R$ and $Y \leftarrow \mathcal{K}$. Note that $H_{\mathcal{K}}$ is symmetric.

Therefore the solution is (24), with the required change of variables. □

In practice, following K-SVD, each update considers only the signals which use the corresponding atom $a$ in their representation, thus $R$ and $\mathcal{K}$ contain only the columns corresponding to these signals.

With regard to Algorithm 2 and Algorithm 3, using the kernel DPL variant requires all the previous modifications implied separately by both DPL and kernel DL variants, respectively. Thus, in Algorithm 2 the error is computed as $E = I - AX = I - AB\mathcal{K}$, and in step 4 the individual $(d, x)$ updates are now based on the $(a, b)$ pair defined in Corollary 7. In Algorithm 3, step 1 will call the kernel OMP$(A^T\tilde{\mathcal{K}}, A^T\mathcal{K}A, s)$, steps 2 and 4 are no longer required, and in step 5 we zero the rows of $\tilde{X}$ where $\|b\mathcal{K}\|$ is (almost) null.

Table 2: Datasets from ODDS used in the experiments

| Dataset | No. features | No. samples | No. outliers | Sparsity | $\beta$ DL | $\beta$, $\gamma$ DPL | |
|---|---|---|---|---|---|---|---|
| satellite | 36 | 6435 | 2036(32%) | 6 | 0.5 | 0.02 | 0.07 |
| shuttle | 9 | 49097 | 3511(7%) | 4 | 0.32 | 0.14 | 0.03 |
| pendigits | 16 | 6870 | 156(2.27%) | 5 | 0.115 | 0.14 | 0.04 |
| speech | 400 | 3686 | 61(1.65%) | 10 | 0.05 | 0.3 | 0.14 |
| mnist | 100 | 7603 | 700(9.2%) | 8 | 0.01 | 0.02 | 0.16 |
| Dataset | No. features | No. samples | No. outliers | Sparsity | $\beta$ KDL | $\beta$, $\gamma$ KDPL | |
| cardio | 21 | 1831 | 176(9.6%) | 5 | 0.15 | 0.05 | 0.05 |
| glass | 9 | 214 | 9(4.2%) | 3 | 0.05 | 0.35 | 0.45 |
| thyroid | 6 | 3772 | 93(2.5%) | 3 | 0.05 | 0.05 | 0.05 |
| vowels | 12 | 1456 | 50(3.4%) | 4 | 0.5 | 0.05 | 0.35 |
| wbc | 30 | 278 | 21(5.6%) | 6 | 0.45 | 0.25 | 0.05 |
| lympho | 18 | 148 | 6(4.1%) | 4 | 0.3 | 0.15 | 0.05 |

## 6. Results

Before proceeding with the numerical simulations, we remind and collect in Table 1 the important hyper-parameters and variables needed during training. The first column lists the algorithms using the parameters and the last column describes their role within the algorithm. The last two rows reflect the DL-OCSVM fusion where $\omega$ and $\lambda$ are used by both DL and OCSVM algorithms. In order to validate our algorithms we performed a set of experiments using the popular Outlier Detection Datasets (ODDS) [1]. The number of features, samples, and outliers (with the associated contamination rate) together with the corresponding sparsity used by our models across the experiments for each of these datasets are presented in Table 2.

We set the maximum number of outer iterations $K = 6$ across all the experiments. A grid-search over 20 randomly generated dictionaries was performed for each of our models with the purpose of finding the dictionary that, together with the other hyperparameters, produces the best Balanced Accuracy (BA), which is defined as the arithmetic mean of True Positive Rate (TPR) and True Negative Rate (TNR). TPR represents the proportion of actual anomalies that were correctly identified as outliers and, respec-

[1] `https://odds.cs.stonybrook.edu/`

tively, TNR the proportion of actual normal samples that were accurately identified as normal.

We compare the results obtained by our methods with the state of the art AD methods One Class - Support Vector Machine (OC-SVM) [4], Local Outlier Factor (LOF) [56], Isolation Forest [57] and Autoencoders (AE) [58] that were optimized through an extensive grid-search across multiple kernels, metrics and hyperparameters. For AE we used symmetric encoders and decoders, each having 3 or 4 layers (depending on the number of features in the datasets) and the same activation function for all layers. The optimizer, dropout rate and activation functions were optimized by performing grid-search for each dataset. In order to emphasize the benefits of combining DL and OC-SVM, we also report results for a baseline DL method and a kernelized one. Both of them use the representation errors in order to detect the outliers, with a threshold that is determined by the real contamination rates of the datasets. The same 20 randomly generated dictionaries are used in the tests. Linear OC-SVM is included with the purpose of highlighting the superior performance of DL-OCSVM and DPL-OCSVM over simple OC-SVM, thereby demonstrating that the modified DL/DPL reprezentations enhance the discrimination capacity of linear OC-SVM. For fairness, when comparing against kernel DL variants we use the best non-linear kernel OC-SVM variant identified through grid-search. Experiments were implemented using Scikit-learn 1.3.0 and Python 3.11.5 on an AMD Ryzen Threadripper PRO 3955WX. Our algorithms implementation and their applications are public and available online at `https://github.com/iulianhiji/AD-DL-OCSVM`.

We start by presenting the convergence of the inner iterations of our DL-OCSVM algorithm in Figure 2(Left), where the dotted lines mark the end of an outer iteration. The total error computed after each inner iteration is strictly decreasing within each outer iteration. Also, after a number of only 4 or 5 outer iterations further computations do not seem to offer significant improvement. In Figure 2(Right) we can see how the total error, DL error (from (9)), OC-SVM error (from (10)) and BA vary across outer iterations. As we can see, even if the total error converges, the OC-SVM error is non-monotonic, replicating the effect on BA. This artifact appears because we use an implementation of OC-SVM which does not support restarting.

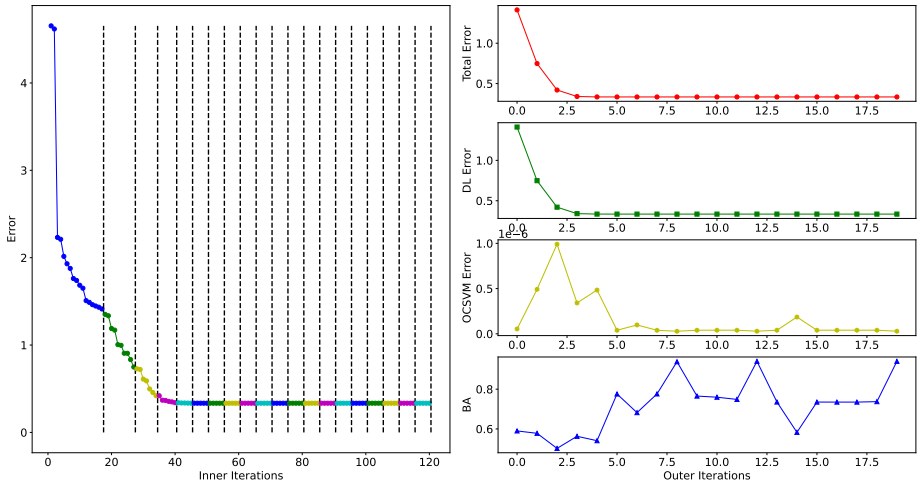


Figure 2: (Left) Convergence on shuttle dataset: each point is an inner iteration, vertical lines separate outer iterations. (Right) Outer iterations convergence analysis of total error $\mathcal{L}$ (first), DL error $F$ (second), OC-SVM error $G$ (third) and BA variation (fourth).

We designed multiple experiments based on the two approaches discussed at the end of Section 3. In the first experiment we used the entire dataset for both training and prediction in order to see how our models behave when they have to analyze a single batch of data. We trained the models described in Sections 4.1 and 4.2 using different values for $\beta$ and $\gamma$ (in the DPL model) in order to find the best ones; the result parameters can be found in the last two columns of Table 2. The BA obtained by our models in this experiment are better than the ones of the competing models in 3 out of the 5 datasets and they closely follow the maximal ones in the other 2 cases, as it can be observed in Table 3. The tables that present the scores of the tested methods use the bold font to emphasize the best one for each dataset. When compared to the proposed method from [19, Table 1], it can be easily observed that the inclusion of OC-SVM in the DL process has brought a clear overall improvement.

For further insights, in Table 4 we show the TPR and TNR corresponding to results from Table 3 while in Table 6 we present the training and testing times obtained while using the entire dataset for both. The displayed times are generally longer than the ones corresponding to the competing models (with a few exceptions) due to the fact tha we

Table 3: Max Balanced accuracy for all data

| Dataset | DL-OCSVM | DPL-OCSVM | DL | OC-SVM | LOF | IForest | AE |
|---|---|---|---|---|---|---|---|
| satellite | 0.7319 | **0.7760** | 0.6324 | 0.3016 | 0.5677 | 0.7105 | 0.6543 |
| shuttle | 0.9400 | 0.9481 | 0.8118 | 0.4757 | 0.5316 | 0.9768 | **0.9782** |
| pendigits | 0.8163 | 0.8724 | 0.5899 | 0.6446 | 0.5895 | **0.8725** | 0.7669 |
| speech | 0.5283 | **0.5915** | 0.5080 | 0.4889 | 0.5 | 0.5232 | 0.5246 |
| mnist | 0.8402 | **0.8528** | 0.6219 | 0.5580 | 0.5736 | 0.7441 | 0.7320 |

Table 4: TPR(left) and TNR(right) for all data

| Dataset | DL-OCSVM | | DPL-OCSVM | | DL | | OC-SVM | | LOF | | IForest | | AE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TPR | TNR | TPR | TNR | TPR | TNR | TPR | TNR | TPR | TNR | TPR | TNR | TPR | TNR |
| satellite | 0.634 | 0.830 | **0.694** | **0.858** | 0.4980 | 0.7667 | 0.5162 | 0.0871 | 0.326 | 0.810 | 0.454 | 0.967 | 0.528 | 0.781 |
| shuttle | 0.886 | 0.995 | 0.904 | 0.993 | 0.6511 | 0.9726 | 0.6539 | 0.2975 | 0.130 | 0.933 | 0.983 | 0.971 | **0.960** | **0.996** |
| pendigits | 0.949 | 0.684 | 0.968 | 0.777 | 0.1987 | 0.9810 | 0.7885 | 0.5007 | 0.263 | 0.916 | **1.0** | **0.745** | 0.545 | 0.989 |
| speech | 0.984 | 0.073 | **0.443** | **0.740** | 0.0328 | 0.9832 | 0.4754 | 0.5023 | 0 | 1.0 | 0.049 | 0.997 | 0.066 | 0.984 |
| mnist | 0.917 | 0.763 | **0.869** | **0.837** | 0.3143 | 0.9295 | 0.7486 | 0.3674 | 0.216 | 0.932 | 0.649 | 0.840 | 0.514 | 0.950 |

perform the OC-SVM training in each outer iteration. Table 6 does not show testing times for LOF because the scikit-learn implementation does not support prediction as a separate action.

Based on the scores that the models provide, we also present a threshold independent score (AUROC) computed in the same scenario in Table 5. Similar to Table 3, our method outperforms the other models for satellite and mnist datasets.

In Table 7 we present the results obtained with our Kernel DL-OCSVM method from Section 5.1. Again, all the data were used both in training and testing. We achieved the best results in 3 out of 6 datasets, proving that our kernel implementation manages to make use of the induced non-linearity in order to detect the outliers and were close to the best result in two other datasets.

Next we focus on the effect of the hyperparameters $\beta$ and $\gamma$ on the anomaly detection process.

Table 5: Max AUROC obtained for all data

| Dataset | DL-OCSVM | DL | OC-SVM | LOF | IForest | AE |
|---|---|---|---|---|---|---|
| satellite | **0.8065** | 0.6691 | 0.6966 | 0.5788 | 0.7194 | 0.6824 |
| shuttle | 0.9134 | 0.9444 | 0.9884 | 0.5557 | **0.9970** | 0.9902 |
| pendigits | 0.8790 | 0.8276 | 0.9513 | 0.5282 | 0.9535 | **0.9680** |
| speech | 0.5286 | 0.5686 | 0.6283 | **0.6921** | 0.4890 | 0.4801 |
| mnist | **0.9101** | 0.7566 | 0.8574 | 0.5965 | 0.8503 | 0.8934 |

Table 6: Training(left) and testing(right) time(in seconds) for all data

| Dataset | DL-OCSVM | | DPL-OCSVM | | DL | | OC-SVM | | LOF | | IForest | | AE | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| satellite | 5.7 | 1.34 | 16.9 | 1.20 | 3.83 | 0.007 | 0.811 | 0.364 | 2.49 | - | 0.73 | 0.11 | 25.69 | 0.31 |
| shuttle | 100.1 | 7.1 | 126.3 | 7.1 | 8.46 | 0.04 | 32.94 | 15.35 | 449.43 | - | 0.64 | 0.30 | 36.17 | 2.59 |
| pendigits | 1.16 | 0.84 | 7.3 | 0.80 | 1.91 | 0.007 | 0.73 | 0.33 | 1.09 | - | 0.90 | 0.11 | 23.77 | 0.40 |
| speech | 21.3 | 7.97 | 129.8 | 3.61 | 30.36 | 0.015 | 1.04 | 0.47 | 1.40 | - | 0.45 | 0.18 | 3.39 | 0.21 |
| mnist | 31.1 | 3.78 | 52.3 | 2.88 | 28.39 | 0.012 | 1.65 | 0.78 | 7.45 | - | 0.53 | 0.18 | 31.21 | 0.39 |

In Figure 3, we show in the left side the mean and the standard deviation of the balanced accuracy obtained in the experiments for different values of $\beta$ while using the 20 randomly initialized dictionaries. In the right side, we plot the maximum balanced accuracy corresponding to the same random dictionaries. For some of the datasets (like satellite, speech and pendigits) both mean and maximum BA have a lower variance, while for others (like mnist and shuttle) both metrics have a higher variance. The same metrics are analyzed for different values of $\beta$ and $\gamma$ from the DPL formulation from Section 4.2. The values corresponding to the lympho dataset are displayed in Figure 4. We can see that we obtain better BA when both $\beta$ and $\gamma$ have larger values.

The next hyperparameter to investigate is the contamination rate and the robustness of our models when we vary the number of outliers found in the dataset. The entire dataset is used for training. We depict our findings in Figure 5 where we show the variation of BA obtained for all the datasets when we use all the inliers and an increasing number of the outliers for training. At the end we test the resulted representations. The models are trained using the optimal DL parameters from Table 2. Because each time

Table 7: Maximum Balanced accuracy obtained for all data on smaller datasets

| Dataset | KDL-OCSVM | KDPL-OCSVM | KDL | KOC-SVM | LOF | IForest | AE |
|---|---|---|---|---|---|---|---|
| cardio | 0.7874 | 0.7888 | 0.7988 | 0.7896 | 0.5564 | **0.8620** | 0.8617 |
| glass | **0.8951** | 0.8292 | 0.5940 | 0.7731 | 0.8566 | 0.7132 | 0.5360 |
| thyroid | 0.7331 | 0.8876 | 0.7186 | 0.7565 | 0.5648 | **0.9506** | 0.7627 |
| vowels | **0.7869** | 0.7843 | 0.7411 | 0.7588 | 0.7419 | 0.6663 | 0.6582 |
| wbc | **0.8991** | 0.8501 | 0.8473 | 0.7703 | 0.8137 | 0.8655 | 0.7969 |
| lympho | 0.9225 | 0.8122 | 0.9096 | 0.8309 | 0.8192 | 0.8873 | **0.9964** |

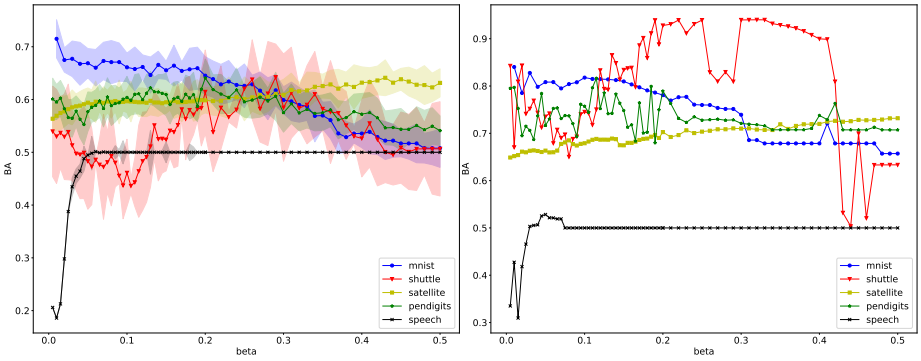


Figure 3: Mean (left) and maximum (right) BA for different values of $\beta$ on multiple datasets.

we start the training process from the beginning with a different number of outliers we obtain variations in the balanced accuracy. This happens because for each set of data a different dictionary, a different set of representations and a different OC-SVM model are produced without a grid-search. This has the side-benefit of depicting again the robustness to hyperparameter values as it can be seen that the ranges of BA contain values that are near the maximal ones from Table 3.

In the last experiment, we want to analyze how our models behave when they have to make predictions on a new dataset different from the one it was trained on. In order to achieve this, we designed an experiment that makes use of KFold cross-validation [59]. The data are split into a validation set (80%) and a testing set (20%). The validation

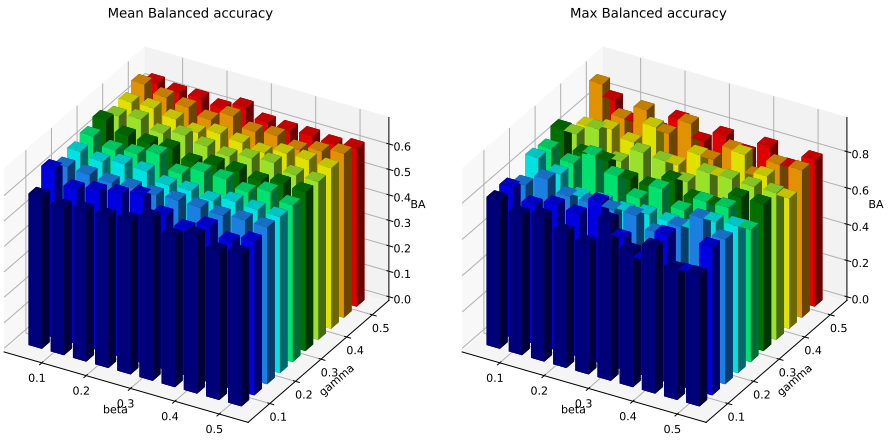


Figure 4: Grid-search for $\beta$ and $\gamma$ on lympho dataset with the resulting BA mean (left) and maximum (right).

Table 8: Balanced accuracy obtained for testing data in the KFold CV scenario

| Dataset | DL-OCSVM | DL | OC-SVM | LOF | IForest | AE |
|---|---|---|---|---|---|---|
| satellite | 0.6354 | 0.5 | 0.3682 | 0.5764 | **0.6979** | 0.6272 |
| shuttle | 0.5164 | 0.7664 | 0.5160 | 0.5174 | **0.9769** | 0.8850 |
| pendigits | **0.8041** | 0.4408 | 0.3389 | 0.5533 | 0.5585 | 0.7436 |
| speech | **0.6196** | 0.5 | 0.4595 | 0.5452 | 0.4924 | 0.4917 |
| mnist | **0.7879** | 0.5137 | 0.2497 | 0.6204 | 0.6186 | 0.6850 |

set is used by the KFold cross-validation procedure to select the best hyperparameters (including one of the 20 randomly generated dictionaries) and then the model is trained on the entire validation set using the found hyperparameters. The reported results are those obtained by making predictions on the testing set. As we can see in Table 8 we outperformed the competing models in 3 out of 5 datasets and come in close with the others.

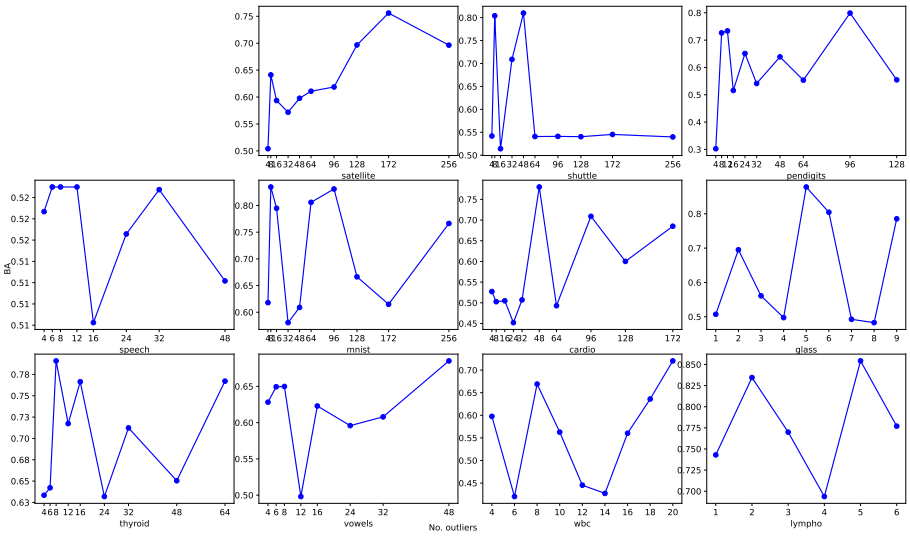


Figure 5: BA obtained in first scenario for different no. of outliers used for training + prediction

## 7. Conclusions and future work

In this paper we proposed DL-OCSVM, a new unsupervised anomaly detection framework, described in Algorithm 1, based on the composite objective of uniform support dictionary learning and one class support vector machines. Our theoretical results focused on modified K-SVD iterations that fuse the uniform support approach with the OC-SVM objective based on standard DL and the DPL formulations. For these we provided Theorems 1 and 5 showing the optimal iteration steps. Plugging the modified K-SVD iterations in the anomaly detection framework produced Algorithms 2 and 3 for model training and, respectively, the anomaly detection routine of new data points. The convergence of the training iterations was analyzed in Proposition 3. Furthermore, we studied in Section 5 the extensions of the existing algorithms and their associated theorems to kernel methods.

For all the proposed algorithms and their extensions we provided numerical analysis and experiments for the models parameters together with performance comparison against standard OC-SVM and uniform support DL and also against Local Outlier Factor and Isolation Forest methods which are often found in the field. Of note is the

comparisons with deep autoencoders which showed that our methods provide similar or better anomaly detection.

In the future we plan to tackle the limitations of our proposed framework such as making it more robust to initialization, reducing computation times and improving the model accuracy. Regarding initialization it is important to mention that the reported results are influenced by the selected random dictionaries. Beyond our current approach, we plan to extend our work to study fusing other popular methods such as Support Vector Data Description, Lightweight Online Detector of Anomalies and Autoencoders.

## Acknowledgments

This work was partially supported by a grant of the Ministry of Research, Innovation and Digitization, CCCDI - UEFISCDI, project number PN-IV-P6-6.3-SOL-2024-2-0197, within PNCDI IV.

## References

[1] C. C. Aggarwal, An introduction to outlier analysis, Springer, 2017.

[2] A. Moujahid, F. Dornaika, Advanced unsupervised learning: a comprehensive overview of multi-view clustering techniques, Artificial Intelligence Review 58 (8), 2025, 234.

[3] B. Dumitrescu, P. Irofti, Dictionary learning algorithms and applications, Springer, 2018.

[4] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, et al., Support vector method for novelty detection., NIPS, Vol. 12, Citeseer, 1999, 582–588.

[5] S. Cai, W. Zuo, L. Zhang, X. Feng, P. Wang, Support vector guided dictionary learning, Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, Springer, 2014, 624–639.

[6] J. Dong, L. Yang, C. Liu, W. Cheng, W. Wang, Support vector machine embedding discriminative dictionary pair learning for pattern classification, Neural Networks 155, 2022, 498–511.

[7] B. Liu, B. Zhou, Y. Xiao, Z. Wang, B. Li, S. He, C. Ye, F. Cao, Smt-dl: A semi-supervised multi-task learning framework based on dictionary learning for robust feature sharing, Neurocomputing, 2025, 130996.

[8] S. Sadik, M. Et-tolba, B. Nsiri, An efficient multivariate approach to dictionary learning for portfolio selection, Digital Signal Processing 153, 2024, 104647.

[9] Z. Xiao, H. Chen, Y. Mi, C. Luo, S.-J. Horng, T. Li, Joint subspace learning and subspace clustering based unsupervised feature selection, Neurocomputing 635, 2025, 129885.

[10] W. Wu, Y. Gu, Advancing unsupervised graph anomaly detection: A multi-level contrastive learning framework to mitigate local consistency deception, Neurocomputing, 2025, 130507.

[11] M. Elad, Sparse and redundant representations: from theory to applications in signal and image processing, Springer Science & Business Media, 2010.

[12] Y. Zhang, D. M. Chandler, M. C. Farias, Motion deblurring via multiscale residual convolutional dictionary learning, Digital Signal Processing, 2025, 105337.

[13] J. Lu, L. Zhang, X. Zhou, M. Li, W. Li, S. Gu, Learned image compression with dictionary-based entropy model, Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, 12850–12859.

[14] C. Rusu, On learning with shift-invariant structures, Digital Signal Processing 99, 2020, 102654.

[15] P. Irofti, L. Romero-Ben, F. Stoican, V. Puig, Learning dictionaries from physical-based interpolation for water network leak localization, IEEE Transactions on Control Systems Technology 32 (3), 2023, 755–766.

[16] Z. Liu, L. Wang, Z. Yin, Y. Xue, Task-driven joint dictionary learning model for multi-view human action recognition, Digital Signal Processing 126, 2022, 103487.

[17] L. Jing, J. Huang, J. Luo, F. Zhenfang, M. Jiancheng, Tunable sparsity fusion sparse coding-driven sparse representation classification for planetary gearbox fault diagnosis, Digital Signal Processing, 2025, 105530.

[18] L. Xiao, Y. Han, X. Ma, M. Dai, The statistical properties analysis of reconstruction error in greedy pursuit algorithms: Taking omp as an example, Electronics Letters 59 (22), 2023, e13023.

[19] P. Irofti, C. Rusu, A. Pătrașcu, Dictionary learning with uniform sparse representations for anomaly detection, ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, 3378–3382.

[20] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, International conference on machine learning, PMLR, 2018, 4393–4402.

[21] M.-N. Nguyen, N. A. Vien, Scalable and interpretable one-class svms with deep learning and random fourier features, Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18, Springer, 2019, 157–172.

[22] Y. Pati, R. Rezaiifar, P. Krishnaprasad, Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition, 27th Asilomar Conf. Signals Systems Computers, Vol. 1, 1993, 40–44.

[23] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE Trans. Signal Proc. 54 (11), 2006, 4311–4322.

[24] Y. Zhou, X. Liang, W. Zhang, L. Zhang, X. Song, Vae-based deep svdd for anomaly detection, Neurocomputing 453, 2021, 131–140.

[25] I. T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences 374 (2065), 2016, 20150202.

[26] K. Pearson, Liii. on lines and planes of closest fit to systems of points in space, The London, Edinburgh, and Dublin philosophical magazine and journal of science 2 (11), 1901, 559–572.

[27] J. Li, H. Zhang, L. Zhang, L. Ma, Hyperspectral anomaly detection by the use of background joint sparse representation, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 8 (6), 2015, 2523–2533.

[28] S. Gu, L. Zhang, W. Zuo, X. Feng, Projective dictionary pair learning for pattern classification, Advances in neural information processing systems 27 (2014).

[29] J. Song, Z. Liu, C. Xie, C. Lu, J. Zhao, S. Gao, Relaxed support vector based dictionary learning for image classification, Multimedia Tools and Applications, 2023, 1–25.

[30] J. Song, X. Xie, G. Shi, W. Dong, Exploiting class-wise coding coefficients: Learning a discriminative dictionary for pattern classification, Neurocomputing 321, 2018, 114–125.

[31] B.-Q. Yang, X.-P. Guan, J.-W. Zhu, C.-C. Gu, K.-J. Wu, J.-J. Xu, Svms multi-class loss feedback based discriminative dictionary learning for image classification, Pattern Recognition 112, 2021, 107690.

[32] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher Discrimination Dictionary Learning for Sparse Representation, IEEE International Conference on Computer Vision, 2011, 543–550.

[33] J. Song, X. Xie, G. Shi, W. Dong, Multi-layer discriminative dictionary learning with locality constraint for image classification, Pattern Recognition 91, 2019, 135–146.

[34] Z. Deng, X. Chen, Y. Xie, Z. Zou, H. Zhang, Multiple structured latent double dictionary pair learning for cross-domain industrial process monitoring, Information Sciences 648, 2023, 119514.

[35] W. Zhu, B. Peng, C. Chen, H. Chen, Deep discriminative dictionary pair learning for image classification, Applied Intelligence, 2023, 1–14.

[36] Z. Chen, X.-J. Wu, T. Xu, J. Kittler, Discriminative dictionary pair learning with scale-constrained structured representation for image classification, IEEE Transactions on Neural Networks and Learning Systems 34 (12), 2023, 10225–10239.

[37] Y. Wang, H. Du, Y. Zhang, Y. Zhang, Efficient and robust discriminant dictionary pair learning for pattern classification, Digital Signal Processing 118, 2021, 103227.

[38] H. Du, Y. Zhang, L. Ma, F. Zhang, Structured discriminant analysis dictionary learning for pattern classification, Knowledge-Based Systems 216, 2021, 106794.

[39] A. Abdi, M. Rahmati, M. Ebadzadeh, Dictionary learning enhancement framework: Learning a non-linear mapping model to enhance discriminative dictionary learning methods, Neurocomputing 357, 2019, 135–150.

[40] J. Dong, K. Wu, C. Liu, X. Mei, W. Wang, Discriminative analysis dictionary learning with adaptively ordinal locality preserving, Neural Networks 165, 2023, 298–309.

[41] S. Amini, M. Sadeghi, M. Joneidi, M. Babaie-Zadeh, C. Jutten, Outlier-aware dictionary learning for sparse representation, 2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2014, 1–6.

[42] P. A. Forero, S. Shafer, J. D. Harguess, Sparsity-driven laplacian-regularized outlier identification for dictionary learning, IEEE Transactions on Signal Processing 65 (14), 2017, 3803–3817.

[43] B. M. Whitaker, D. V. Anderson, Learning anomalous features via sparse coding using matrix norms, 2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE), IEEE, 2015, 196–201.

[44] M. Bevilacqua, S. Tsaftaris, Dictionary-decomposition-based one-class svm for unsupervised detection of anomalous time series, Proceedings of 23rd European signal processing conference (EUSIPCO), 2015, 1776–1780.

[45] J. He, Z. Cheng, B. Guo, Anomaly detection in telemetry data using a jointly optimal one-class support vector machine with dictionary learning, Reliability Engineering & System Safety 242, 2024, 109717.

[46] Z. Cheng, S. Wang, P. Zhang, S. Wang, X. Liu, E. Zhu, Improved autoencoder for unsupervised anomaly detection, International Journal of Intelligent Systems 36 (12), 2021, 7103–7125.

[47] B. Liu, X. Li, Y. Xiao, P. Sun, S. Zhao, T. Peng, Z. Zheng, Y. Huang, Adaboost-based svdd for anomaly detection with dictionary learning, Expert Systems with Applications 238, 2024, 121770.

[48] S. Lin, M. Zhang, X. Cheng, K. Zhou, S. Zhao, H. Wang, Dual collaborative constraints regularized low-rank and sparse representation via robust dictionaries construction for hyperspectral anomaly detection, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 16, 2022, 2009–2024.

[49] X. Hu, Z. Li, L. Luo, H. R. Karimi, D. Zhang, Dictionary trained attention constrained low rank and sparse autoencoder for hyperspectral anomaly detection, Neural Networks, 2024, 106797.

[50] A. Ben-Tal, M. Teboulle, Hidden convexity in some nonconvex quadratically constrained quadratic programming, Mathematical Programming 72 (1), 1996, 51–63.

[51] J. J. Moré, D. C. Sorensen, Computing a trust region step, SIAM Journal on scientific and statistical computing 4 (3), 1983, 553–572.

[52] C. Fortin, H. Wolkowicz, The trust region subproblem and semidefinite programming, Optimization methods and software 19 (1), 2004, 41–67.

[53] E. Hazan, T. Koren, A linear-time algorithm for trust region problems, Mathematical Programming 158 (1), 2016, 363–381.

[54] J. Sun, On piecewise quadratic newton and trust region problems, Mathematical programming 76 (3), 1997, 451–467.

[55] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, R. Chellappa, Design of non-linear kernel dictionaries for object recognition, IEEE Transactions on Image Processing 22 (12), 2013, 5123–5135.

[56] M. Breunig, H. Kriegel, R. Ng, J. Sander, LOF: identifying density-based local outliers, Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, 93–104.

[57] T. Fei, M. Kai, K. M. Ting, Z. Zhi-Hua, Isolation forest, 2008 Eighth IEEE International Conference on Data Mining, 2008, 413–422.

[58] M. A. Kramer, Autoassociative neural networks, Computers & chemical engineering 16 (4), 1992, 313–328.

[59] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, S. Ridella, et al., The'k'in k-fold cross validation., ESANN, Vol. 102, 2012, 441–446.

**Supplementary Material: Algorithms**

Algorithms 2 and 3 refer to the standard DL setting. For completeness, we provide below the corresponding algorithms for the DPL, kernel DL and kernel DPL alternatives. For the DPL formulation described in Section 4.2, the Algorithms are 4 and 5. For the kernel DL formulation in Section 5.1, Algorithms 6 and 7. For the kernel DPL in section 5.2, Algorithms 8 and 9.

**Algorithm 4:** DPL-OCSVM Uniform Representation Learning

**Data:** train set $Y \in \mathbf{R}^{m\times N}$, $D^{k-1} \in \mathbf{R}^{m\times n}$, $P^{k-1} \in \mathbf{R}^{n\times m}$,
inner iteration k

**Result:** dictionary $D^k$ and projection matrix $P^k$

Error: $E^k = Y - D^{k-1}P^{k-1}Y$

**for** $i \in \{1,\dots,n\}$ **do**

  Atom error: $R^k = E^k + d_i^{k-1}p^{i,k-1}Y$

  Update: new $(d_i^k, p^{i,k})$ according to Theorem 5

  New error: $E^k = R^k - d_i^k p^{i,k}Y$

**Algorithm 5:** DPL-OCSVM Anomaly Detection

**Data:** test set $\tilde{Y} \in \mathbf{R}^{m\times\tilde{N}}$, dictionary $D \in \mathbf{R}^{m\times n}$,
projection matrix $P \in \mathbf{R}^{n\times m}$, sparsity $s$, tolerance $tol$ and
OC-SVM model $(\omega, \rho, \lambda)$

**Result:** anomalies $\tilde{\mathcal{A}}$

Representation: $\tilde{X} = \mathrm{OMP}(\tilde{Y}, D, s)$

**for** $i \in \{1,\dots,n\}$ **do**

  Trimming: zero $\tilde{x}^i$ if $\|p^i\tilde{Y}\| < tol$

**if** $sign\left(\omega^T\tilde{x}_i - \rho\right) \le 0$ **then** $\tilde{\mathcal{A}} = \tilde{\mathcal{A}} \cup \{i\} \quad \forall i \in \tilde{N}$

**Algorithm 6:** KDL-OCSVM Uniform Representation Learning

**Data:** Gram matrix $\mathcal{K} \in \mathbf{R}^{N \times N}$, $A^{k-1} \in \mathbf{R}^{N \times n}$, inner iteration k

**Result:** dictionary $A^k$ and representations $X^k$

Error: $E^k = I - A^{k-1}X^{k-1}$

**for** $i \in \{1, \ldots, n\}$ **do**

  Atom error: $R^k = E^k + a_i^{k-1} x^{i,k-1}$

  Update: new $(a_i^k, x^{i,k})$ according to Corollary 6

  New error: $E^k = R^k - a_i^k x^{i,k}$

**Algorithm 7:** KDL-OCSVM Anomaly Detection

**Data:** Gram matrices $\tilde{\mathcal{K}} \in \mathbf{R}^{N \times \tilde{N}}$ and $\mathcal{K} \in \mathbf{R}^{N \times N}$, dictionary $A \in \mathbf{R}^{N \times n}$, sparsity $s$, and OC-SVM model $(\omega, \rho, \lambda)$

**Result:** anomalies $\tilde{\mathcal{A}}$

Representation: $\tilde{X} = \mathrm{OMP}(A^T\tilde{\mathcal{K}}, A^T\mathcal{K}A, s)$

Error: $E = \mathcal{K}^{-1}\tilde{\mathcal{K}} - A\tilde{X}$

**for** $i \in \{1, \ldots, n\}$ **do**

  Atom error: $R = E + d_i \tilde{x}^i$

  Trimming: zero $\tilde{x}^i$ if condition from (5) holds

**if** $sign\left(\omega^T \tilde{x}_i - \rho\right) \leq 0$ **then** $\tilde{\mathcal{A}} = \tilde{\mathcal{A}} \cup \{i\} \quad \forall i \in \tilde{N}$

**Algorithm 8:** KDPL-OCSVM Uniform Representation Learning

**Data:** Gram matrix $\mathcal{K} \in \mathbf{R}^{N \times N}$, $A^{k-1} \in \mathbf{R}^{N \times n}$, inner iteration k

**Result:** dictionary $A^k$ and projection matrix $B^k$

Error: $E^k = I - A^{k-1}B^{k-1}\mathcal{K}$

**for** $i \in \{1, \ldots, n\}$ **do**

  Atom error: $R^k = E^k + a_i^{k-1} b^{i,k-1}\mathcal{K}$

  Update: new $(a_i^k, b^{i,k})$ according to Corollary 7

  New error: $E^k = R^k - a_i^k b^{i,k}\mathcal{K}$

**Algorithm 9:** KDPL-OCSVM Anomaly Detection

**Data:** Gram matrices $\tilde{\mathcal{K}} \in \mathbf{R}^{N \times \tilde{N}}$ and $\mathcal{K} \in \mathbf{R}^{N \times N}$, dictionarie $A \in \mathbf{R}^{N \times n}$, projection matrix $B \in \mathbf{R}^{n \times N}$, sparsity $s$, tolerance $tol$ and OC-SVM model $(\omega, \rho, \lambda)$

**Result:** anomalies $\tilde{\mathcal{A}}$

Representation: $\tilde{X} = \text{OMP}(A^T\tilde{\mathcal{K}}, A^T\mathcal{K}A, s)$

**for** $i \in \{1, \ldots, n\}$ **do**

    Trimming: zero $\tilde{x}^i$ if row $\|b^i\tilde{\mathcal{K}}\| < tol$ is null

**if** $sign\left(\omega^T\tilde{x}_i - \rho\right) \leq 0$ **then** $\tilde{\mathcal{A}} = \tilde{\mathcal{A}} \cup \{i\} \quad \forall i \in \tilde{N}$