Support Vector Data Description (SVDD)
An experimental study for Anomaly detection

Ezukwoke K.I[1], Zareian S.J[2]

[1, 2] Department of Computer Science
Machine Learning and Data Mining
{ifeanyi.ezukwoke, samaneh.zareian.jahromi}@etu.univ-st-etienne.fr
University Jean Monnet, Saint-Etienne, France

Abstract

Support Vector Data Description (SVDD) is a variant of Support Vector Machines (SVM) used for one class classification. It is particularly designed for outlier detection and hence the focus of our paper. In this paper we solve the SVDD optimization problem using gradient descent (primal problem) and minibatch gradient ascent(dual problem). We compare its performance with that of stochastic Support vector Machine (SVM), and test the algorithm on a generated synthetic dataset using kernel and multiple kernel methods.

Keywords
One class classification, support vector classifier, support vector data description (SVDD), outlier detection, novelty detection.

1 INTRODUCTION

Support Vector Data Description (SVDD) is a variant of Support Vector Machines (SVM), usually referred to as the **One class SVM**. It is interesting for use cases where researchers are only interested in the positive class (*class of interest*), therefore making it suitable to detect novel data or outliers [1]. Its objective is to find a hypersphere of the positive class among the remaining classes. It particularly finds use in applications such as anomaly detection [2, 3, 4, 5, 6] and novelty detection [7, 8, 9].

The SVDD model is trained on the target class only and is assumed to understand the boundary (*hypersphere*) of the target class to accurately label the outliers. We begin by introducing the normal formulation of SVDD (primal) then proceed to describe SVDD with error before kernelizing it.

2 Support Vector Data Description (SVDD)

Support Vector Data Description (SVDD), originally proposed by [1] is a model which aims at identifying a spherically shaped boundary around a dataset. This dataset is the target dataset for which we expect the model to find outliers for during testing or deployment. As a result SVDD can be regarded as a description of the class of interest [10].

2.1 Primal and Dual Formulation of SVDD

Given a set of training samples $\{\mathbf{x}_i\}$ where $\mathbf{x} \in \mathcal{X}$ is the feature space, SVDD aims to minimize the radius of a hypershere $\mathbf{R}$ in a linear space subject to the contraint $\|x_i - c\|^2 \leq R^2$ (c is the center of the hypersphere). We can write this as an optimization problem

$$\begin{cases} \underset{R\in\mathbb{R}, c\in\mathbb{R}^d}{min} & R^2 \\ s.t & \|x_i - c\|^2 \leq R^2, i = 1, \dots, n \end{cases} \tag{1}$$

since $\mathbf{w} = \mathbf{c}$ we can rewrite equation (1) as

$$\begin{cases} \underset{R\in\mathbb{R}, c\in\mathbb{R}^d}{min} & R^2 \\ s.t & \|x_i - w\|^2 \leq R^2, i = 1, \dots, n \end{cases} \tag{2}$$

We can also prove that this optimization problem can be rewritten as Quadratic problem given by

$$\begin{cases} \underset{\mathbf{w},\rho}{min} & \frac{1}{2}\|\mathbf{w}\|^2 - \rho \\ s.t & \mathbf{w}^T\mathbf{x} \geq \rho + \frac{1}{2}\|\mathbf{x}\|^2 \end{cases} \tag{3}$$

where $\rho = \frac{1}{2}\left(\|\mathbf{c}\|^2 - R^2\right)$ and $\mathbf{w} = \mathbf{c}$
Proof:

$$\|x_i - c\|^2 \leq R^2 \tag{4}$$
$$\|x_i\|^2 - 2x_i^T c + \|c\|^2 \leq R^2 \tag{5}$$
$$-2x_i^T c \leq R^2 - \|x_i\|^2 - \|c\|^2 \tag{6}$$
$$x_i^T c \geq \underbrace{\frac{1}{2}(\|c\|^2 - R^2)}_{\rho} + \frac{1}{2}\|x_i\|^2 \tag{7}$$

We can solve the optimization problem from equation (2) by introducing Lagrangian multipliers.

$$\mathcal{L}(\mathbf{w}, R, \alpha) = R^2 + \sum_{i=1}^{N} \alpha_i(\|x_i - \mathbf{w}\|^2 - R^2) \tag{8}$$

We recall the Karush-Kuhn-Tucker (**KKT**) optimality conditions as

$$\nabla_R \mathcal{L} = 0 \tag{9}$$
$$\nabla_w \mathcal{L} = 0 \tag{10}$$

$$\nabla_R \mathcal{L} = 2R - 2\alpha R = 0 \tag{11}$$
$$\alpha = 1 \tag{12}$$

$$\nabla_w \mathcal{L} = -2\sum_{i=1}^{N} \alpha_i x_i + 2\sum_{i=1}^{N} w\alpha_i = 0 \tag{13}$$
$$w = \frac{\sum_{i=1}^{N} \alpha_i x_i}{\sum_{i=1}^{N} \alpha_i} \tag{14}$$

From **Representer theorem**, we know that $w = \alpha x$, therefore

$$w = \frac{\sum_{i=1}^{N} \alpha_i x_i}{\sum_{i=1}^{N} \alpha_i} = \sum_{i=1}^{N} \alpha_i x_i \tag{15}$$

If we substitute back the solutions of equations (12) and (15) into (8) we have that

$$\mathcal{L}(\alpha) = \sum_{i=1}^{N} \alpha_i(\|x_i - \sum_{i=1}^{N} \alpha_j x_j\|^2) \tag{16}$$

$$= \alpha(x - \sum_{i=1}^{N} \alpha_j x_j)^T(x - \sum_{i=1}^{N} \alpha_j x_j) \tag{17}$$

$$= \alpha(x^T x - 2\sum_{i=1}^{N} \alpha_i x^T x + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j \alpha x^T x) \tag{18}$$

$$= \alpha x^T x - 2\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j x^T x + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j x^T x \tag{19}$$

$$\mathcal{L}(\alpha) = \sum_{i=1}^{N} \alpha_i x^T x - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j x^T x \tag{20}$$

The Wolfe Dual formulation of SVDD can now be written as a maximization problem

$$\begin{cases} \underset{\alpha\in\mathbb{R}^N}{max} & \alpha\mathbf{diag}(\mathbf{G}) - \alpha_i\mathbf{G}\alpha_j \\ s.t & \mathbf{e}^T\alpha = 1 \\ s.t & 0 \leq \alpha_i \quad 1 = 1, \dots, N \end{cases} \tag{21}$$

Where $G = x^T x$. We maximize the objective function of the dual formulation along the α using gradient ascent algorithm. To do that we need to find the first order derivative for the above

$$\nabla_\alpha \mathcal{L} = diag(G) - \sum_{i=1}^{N} \alpha_i G \tag{22}$$

The gradient ascent algorithm for SVDD without error

Algorithm 1: SVDD without error constraint using **Gradient ascent**

```
Input  : $\mathbf{X}, \alpha$
Output : $\alpha$
1 begin
2     $\alpha_j \leftarrow \alpha^{1}$';
3     while not converged do
4         $\alpha_{j+1} = \alpha_j + lr\nabla_\alpha L$;
5         if $\alpha_{j+1} \leq 0$ then
6             $\alpha_{j+1} \leftarrow 0$
7         end
8     end
9 end
```

Input α is a randomly generated number with sum equal to 1. lr is the learning rate. The support vectors those that satisfy the constraint $0 \leq \alpha \leq C$ and we can calculate the radius of the hypershpere R from the support vectors using $R^2 = \|x_i - c\|^2$.

2.2 Primal and Dual Formulation of SVDD with errors

The primal formulation of SVDD from equation (1) is a strict formulation (hard margin) without regards to data points that lie on the decision boundary. We consider the case where the training data cannot be seperated without errors [11]. By allowing a permissible error $\xi \geq 0$, we establish a soft-margin classifier as seen in Support Vector Machine (SVM)-soft margin classifier.

We rewrite equation (2) with errors as

$$\begin{cases} \underset{R\in\mathbb{R}, c\in\mathbb{R}^d}{min} & R^2 + C\sum_{i=1}^N \xi_i \\ s.t & \|x_i - w\|^2 \leq R^2 + \xi_i, i = 1, \dots, n \\ & \xi_i \geq 0 \quad i = 1, \dots, n \end{cases} \tag{23}$$

Where ξ_i is the vector of slack variables and $C \geq 0$ is the parameter that controls the tradeoff between the volume of the hypersphere and the permitted errors [10].

We rewrite equation (23) as a quadratic problem

$$\begin{cases} \underset{\mathbf{w},\rho}{min} & \frac{1}{2}\|\mathbf{w}\|^2 - \rho + \frac{C}{2}\sum_{i=1}^N \xi_i \\ s.t & \mathbf{w}^T\mathbf{x}_i \geq \rho + \frac{1}{2}\|\mathbf{x}_i\|^2 - \frac{1}{2}\xi_i \\ & \xi_i \geq 0 \quad i = 1, \dots, n \end{cases} \tag{24}$$

where $\rho = \frac{1}{2}\left(\|\mathbf{c}\|^2 - R^2\right)$ We follow similar step used for solving the SVDD without error.

We introduce the Lagrangian multiplies into equation (23)

$$\mathcal{L}(\mathbf{w}, R, \alpha, \xi) = R^2 + \sum_{i=1}^N \alpha_i(\|x_i - \mathbf{w}\|^2 - R^2 - \xi_i)$$

KKT conditions:

$$\nabla_w \mathcal{L} = 0 \tag{25}$$
$$\nabla_R \mathcal{L} = 0 \tag{26}$$
$$\nabla_\alpha \mathcal{L} = 0 \tag{27}$$
$$\nabla_\xi \mathcal{L} = 0 \tag{28}$$

The solution is similar to that of equation (20). So that the maximization problem of dual SVDD with errors becomes,

$$\begin{cases} \underset{\alpha\in\mathbb{R}^N}{max} & \alpha\mathbf{diag}(\mathbf{G}) - \alpha_i\mathbf{G}\alpha_j \\ s.t & \mathbf{e}^T\alpha = 1 \\ & 0 \leq \alpha_i \leq C \quad 1 = 1, \dots, N \end{cases} \tag{29}$$

Where $G = x^Tx$ and $C \leq 1$. We maximize the objective function of the dual formulation along the α.

Algorithm 2: SVDD without error constraint using **Gradient ascent**

```
   Input  : $\kappa, \alpha_j, C$
   Output : $\alpha$
1  begin
2      $\alpha_j \leftarrow \alpha^{1}$';
3      while not converged do
4          $\alpha_{j+1} = \alpha_j + lr\nabla_\alpha L$;
5          if $\alpha_{j+1} \leq 0$ then
6              $\alpha_{j+1} \leftarrow 0$
7          end
8          if $\alpha_{j+1} \geq C$ then
9              $\alpha_{j+1} \leftarrow C$
10         end
11     end
12 end
```

2.3 Kernel Formulation

Given a set of training points $\{x_i \forall i = 1, \dots N\}$ with a feature vector $\mathbf{x}_i \in \mathbb{R}^N$, we map x_i to a non-linear space $\phi(x_i) \in \mathcal{H}$. Where Hilbert space $\mathcal{H}$ is a non-linear kernel space [13]. $\phi(x_i)$ can also be represented as $\kappa(x_i, \cdot)$.

2.3.1 Primal & Dual Formulation without errors

The primal optimization problem for the kernel SVDD can be extended from equation (1) as

$$\begin{cases} \underset{R\in\mathbb{R}, c\in\mathbb{R}^d}{min} \quad R^2 \\ s.t \quad \|\kappa(x_i, \cdot) - w\|^2 \leq R^2, i = 1, \dots, n \end{cases} \tag{30}$$

We introduce the Lagrangian multipliers to the above optimization problem

$$\mathcal{L}(\mathbf{w}, R, \alpha) = R^2 + \sum_{i=1}^{N} \alpha_i(\|\phi(x_i) - \mathbf{w}\|^2 - R^2)$$

Applying the **KKT** conditions we obtain

$$\begin{cases} \nabla_R \mathcal{L}, & \alpha = 1 \\ \nabla_w \mathcal{L}, & \mathbf{w} = \sum_{i=1}^N \alpha_i x_i \end{cases} \tag{31}$$

So that

$$\mathcal{L}(\alpha) = \sum_{i=1}^{N} \alpha_i(\|\phi(x_i) - \sum_{j=1}^{N} \alpha_j x_j\|^2) \tag{32}$$

$$= \alpha \left(\phi(x_i) - \sum_{j=1}^{N} \alpha_j x_j\right)^T \left(x - \sum_{j=1}^{N} \alpha_j \phi(x_i)\right) \tag{33}$$

$$= \alpha(\kappa - 2\sum_{i=1}^{N} \alpha_i \kappa + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \kappa) \tag{34}$$

where $\kappa = \phi(x_i)^T \phi(x_j)$ is a kernel function.

$$= \alpha\kappa - 2\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \kappa + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \alpha \kappa \tag{35}$$

$$\mathcal{L}(\alpha) = \sum_{i=1}^{N} \alpha_i \kappa - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \kappa \tag{36}$$

The dual optimization problem is now given as

$$\begin{cases} \underset{\alpha\in\mathbb{R}^N}{max} \quad \alpha diag(\kappa) - \alpha^T \kappa \alpha \\ s.t \quad \mathbf{e}^T \alpha = 1 \\ \qquad 0 \leq \alpha_i \quad 1 = 1, \dots, N \end{cases} \tag{37}$$

Algorithm 3: Kernel SVDD (hard margin) using **Minibatch Gradient Ascent**

```
   Input  : $\kappa, \alpha_j$
   Output : $\alpha$
1  begin
2      $\alpha_j \leftarrow \alpha^{1}$';
3      while not converged do
4          for
             $i \in$ randshuffle($\{1, \dots, N\}$)
             do
5              $\alpha_{j+1} = \alpha_j + lr\nabla_\alpha L_i$;
6              if $\alpha_{j+1} \leq 0$ then
7                  $\alpha_{j+1} \leftarrow 0$
8              end
9          end
10     end
11 end
```

The support vectors satisfy the constrain $0 \leq \alpha$ and used to calculate the radius of the hypershpere.

2.3.2 Primal & Dual Formulation with errors

Similarly, the soft-margin SVDD follows from the error formulation in equation (23) such that we transform and map x_i to a non-linear space $\phi(x_i)$. This formulation is given by

$$\begin{cases} \underset{R\in\mathbb{R},c\in\mathbb{R}^d}{min} & R^2 + C\sum_{i=1}^{N}\xi \\ s.t & \|\phi(x_i) - w\|^2 \leq R^2 + \xi, i = 1, \ldots, n \\ & \xi \geq 0 \quad i = 1, \ldots, n \end{cases} \tag{38}$$

By introducing Lagrangian multipliers, we have that

$$\mathcal{L}(\mathbf{w}, R, \alpha, \xi) = R^2 + \sum_{i=1}^{N} \alpha_i(\|\phi(x_i) - \mathbf{w}\|^2 - R^2 - \xi)$$

From applying the **KKT** conditions, we know that,

$$\begin{cases} \nabla_R\mathcal{L}, & \alpha = 1 \\ \nabla_w\mathcal{L}, & \mathbf{w} = \alpha x \\ \nabla_\xi\mathcal{L}, & \alpha = 0 \end{cases} \tag{39}$$

Substituting back the optimality result of KKT conditions gives us the Wolfe-Dual the maximization problem

$$\begin{cases} \underset{\alpha\in\mathbb{R}^N}{max} & \alpha\mathbf{diag}(\kappa) - \alpha^T\kappa\alpha \\ s.t & \mathbf{e}^T\alpha = 1 \\ & 0 \leq \alpha_i \leq C \quad 1 = 1, \ldots, N \end{cases} \tag{40}$$

Data samples that satisfy the constraint $0 \leq \alpha \leq C$ are known as the data descriptors (**support vectors**) and they fall on the boundary of the hypersphere. data samples with $\alpha = C$ reside outside the boundary of the hypershere while samples with $\alpha = 0$ lie inside the boundary.

We solve this problem using minibatch gradient ascent algorithm by taking the first order derivative of $\nabla_\alpha\mathcal{L} = diag(\kappa(x_i, x_j)_k) - \alpha\kappa(x_i, x_j)$.

Algorithm 4: Kernel SVDD (soft margin) using **Minibatch Gradient Ascent**

```
    Input  : $\kappa, \alpha_j$
    Output : $\alpha$
1   begin
2       $\alpha_j \leftarrow \alpha^{1}$';
3       while not converged do
4           for $i \in$ randshuffle($\{1, \ldots, N\}$) do
5               $\alpha_{j+1} = \alpha_j + lr\nabla_\alpha L_i$;
6               if $\alpha \leq 0$ then
7                   $\alpha \leftarrow 0$
8               end
9               if $\alpha_{j+1} \geq C$ then
10                  $\alpha_{j+1} \leftarrow C$
11              end
12          end
13      end
14      return $\alpha$
15  end
```

The decision function for SVDD is $f(x) = sign(2\alpha\kappa(x_i, x_j) + b)$, where b is the center of the hypersphere and is given by $R^2 - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j\kappa(x_i, x_j)$.

2.4 Kernels

We introduce the commonly used kernels and a brief overview of Multiple kernels used. The radius R is computed from the

- **Linear kernel**

$$\kappa(x_i, x_j) = \mathbf{x}_i\mathbf{x}_j^T \tag{41}$$

- **Polynomial kernel**

$$\kappa(x_i, x_j) = (\mathbf{x}_i\mathbf{x}_j^T + c)^d \tag{42}$$

  where $c \geq 0$ and d is the degree of the polynomial usually greater than 2.

- **RBF(Radial Basis Function) kernel**

Sometimes referred to as the **Gaussian kernel**.

$$\kappa(x_i, x_j) = \exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2) \quad (43)$$

where $\gamma = \frac{1}{2\sigma^2}$.

- **Sigmoid kernel**

$$\kappa(x_i, x_j) = \tanh(\gamma \mathbf{x}_i \mathbf{x}_j^T + c) \quad (44)$$

where $c \geq 0$ and $\gamma = \frac{1}{2\sigma^2}$.

- **Laplace kernel**

$$\kappa(x_i, x_j) = \exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||) \quad (45)$$

where $\gamma = \frac{1}{2\sigma^2}$.

2.5 Multi-kernel

The reason behind the use of multiple kernel is similar to the notion of multiclassification, where cross-validation is used to select the best performing classifier [14]. By using multiple kernel, we hope to learn a different similar in the kernel space not easily observed when using single kernel.

We can prove from **Mercer's Theorem** that a kernel is **Positive Semi-Definite (PSD)** if $u^T \kappa(x_i, x_j) u \geq 0$. Hence by performing arithmetic or any mathematical operation on two or more kernel matrix, we obtain a new kernel capable of exploiting different property or similarities of training data.

Given a kernel κ, we prove that κ is PD if

$$\langle u, \kappa u \rangle \geq 0 \quad (46)$$

Proposition: *A symmetric function κ: $\chi \to \mathbb{R}$ is positive semi-definite if and only if* $\langle u, \kappa u \rangle \geq 0$
Proof:
Suppose that κ is a kernel which is the inner product of the mapping functions $\langle \phi(x_i)\phi(x_j) \rangle$. κ is a kernel if its inner product are positive and the solution of $\kappa u = \lambda u$ gives non-negative eigenvalues.

So that,

$$\langle u, \kappa u \rangle = \sum_{i=1}^{N} u_i \cdot (\kappa u_i) \quad (47)$$

$$= \sum_{i=1}^{N} u_i \sum_{j=1}^{N} \langle \phi(x_i)\phi(x_j)_{\mathcal{H}} \rangle u_j \quad (48)$$

Where $\mathcal{H}$ represent the Hilbert space we project the kernel [13].

$$= \left\langle \sum_{i=1}^{N} \sum_{j=1}^{N} u_i \phi(x_i), u_j \phi(x_j) \right\rangle_{\mathcal{H}} \quad (49)$$

$$\langle u, \kappa u \rangle = \left\| \sum_{i=1}^{N} u_i \phi(x_i) \right\|_{\mathcal{H}}^2 \geq 0 \quad (50)$$

Therefore κ is positive definite.

Using this property of the kernel κ we introduce multiple kernel combination as follow

- **LinearRBF**
Here we combine two kernels, precisely **Linear and RBF kernel** using their inner product.

$$\hat{\mathbf{K}}_{\mathbf{linrbf}} = \kappa(x_i, x_j) \times \kappa(x_i, x_l) \quad (51)$$

- **RBFPoly**
Here we combine **RBF and Polynomial kernel** using their inner product.

$$\hat{\mathbf{K}}_{\mathbf{rbfpoly}} = \kappa(x_i, x_j) \times \kappa(x_i, x_l) \quad (52)$$

- **EtaKernel**
The **EtaKernel** is a composite combination of **LinearRBF, RBFPoly and RBFCosine** and it is given by

$$\hat{\mathbf{K}}_{\mathbf{etarbf}} = \hat{\mathbf{K}}_{\mathbf{linrbf}} \times \hat{\mathbf{K}}_{\mathbf{rbfpoly}} + \hat{\mathbf{K}}_{\mathbf{rbfpoly}} \times \hat{\mathbf{K}}_{\mathbf{rbfcosine}}$$

3 EXPERIMENT

3.1 Dataset

We use a synthetic dataset, generated specifically for testing the performance of SVDD. We call this dataset **Anomaly dataset**. Using a specific tailored function we generate a dataset consisting of 1000 datapoints with 950 inliers and 50 outliers (imbalanced dataset where the majority class is the class of interest.)

3.2 Performance analysis

Dealing with an imbalanced dataset suffice the use of an evaluation metric that takes into consideration, the *true positive rate* and *false positive rate*. **AUCROC** (Area Under Curve-Receiver Operating Characteristics) score is a classification metric for imbalanced dataset, taking into account both true and false positive rate. The higher the AUCROC score, the better the classic or kernel method.

We also report metrics including *accuracy and f1-score* for each kernel. High score indicate a good model and low score indicate otherwise.

3.2.1 Evaluation metric

- AUCROC Score
  Area Under Curve-Receiver Operating Characteristics score is the performance of all classification thresholds. It is given by the area under the curve of true positive rate against false positive rate. Where **True Positive Rate** corresponds to the proportion of positive data points that are correctly considered as positive with respect to all positive data, and is given by

$$TPR = \frac{TP}{TP + FN} \tag{53}$$

  and **False Positive Rate** corresponds to the proportion of negative data points that are misclassified as positives with respect to all negative data points and is given by

$$FPR = \frac{FP}{FP + TN} \tag{54}$$

- Accuracy
  Accuracy is the measure of how often our classifier makes correct prediction, and it is given by

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{55}$$

- F1-Score
  F1-Score is the harmonic mean of precision and recall and it is given by

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \tag{56}$$

  where precision is given by

$$precision = \frac{TP}{TP + FP} \tag{57}$$

$$recall = \frac{TP}{TP + FN} \tag{58}$$

TP: True Positives, **TN**: True Negatives, **FP**: False Positives **FP**: False Negatives.

3.2.2 Comparing SVDD (soft and hard margin) with SVM (soft and hard margin)

We experiment with $C = 0.01$, Gamma ($\gamma = 1$), Polynomial degree ($d = 5$) and visualize the result for all algorithms on our dataset (figure 1). We observe that linear SVDD performs better in hyperspheing the target class. When compared to SVM using the columns of table 1. SVDD(hard or no error) and SVDD(soft or with error) perform better than linear SVM with AUCROC scores of 88%. We observe the columns of table 1; SVM **rbf, laplace, rbfpoly ,linrbf and etakernel kernels** outperformed SVDD kernels, while SVDD **polynomial and sigmoid kernels** outperformed that of SVM.

Table 2 is a table of accuracy/f1-score for SVDD and SVM on anomaly dataset. Although we prefer to use the AUCROC score, accuracy and f1-score help in understanding the best performing and worst performing algorithms. For instance, we observe zero f1-score for linear soft-margin SVM. This indicate all data points were classified as outliers.

Increasing the value of $C = 0.1$, $\gamma = 5$ and $d = 10$, we observe a noticeable change in the performance of both SVDD and SVM algorithms. AUCROC score of both linear SVDD and SVM is almost unchanged. The performance of all kernels for SVDD and SVM however decreased significantly.

SVDD **rbf, laplace, rbfpoly, linrbf and etakernel** outperformed SVDD (soft and hard margin) kernels. SVDD (**polynomial and sigmoid**) kernels outpeformed those of SVM.

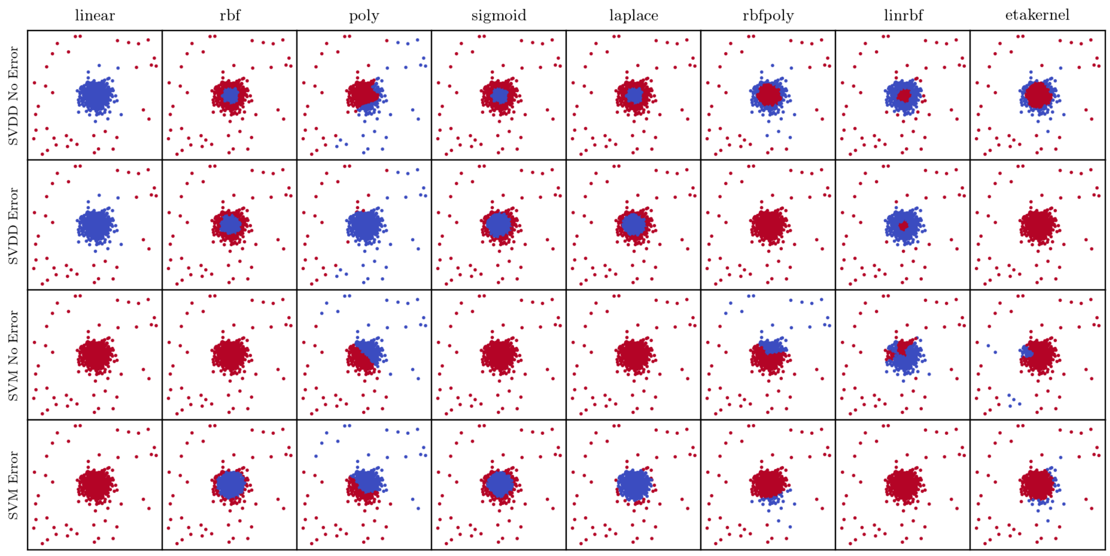

Figure 1: Anomaly detection using SVDD (soft and hard margin) and SVM(soft and hard margin). Both algorithm are trained using $C = 0.01$ $\gamma = 1$ and $d = 5$

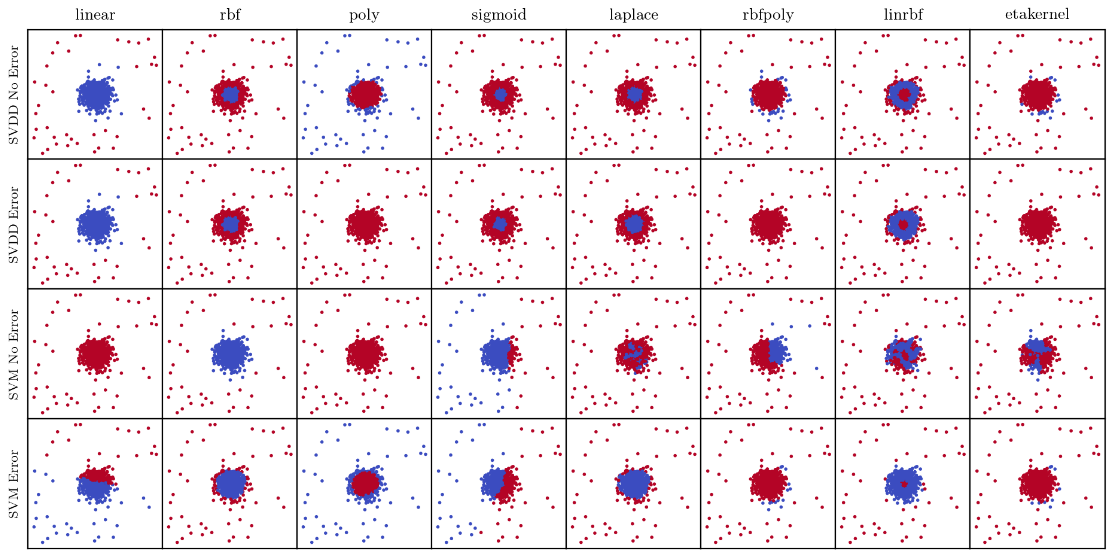

Figure 2: Anomaly detection using SVDD (soft and hard margin) and SVM(soft and hard margin). Both algorithm are trained using $C = 0.1$ $\gamma = 5$ and $d = 10$

AUCROC Score (%)								
kernels	linear	rbf	poly	sigmoid	laplace	rbfpoly	linrbf	eta kernel
SVDD(*no error*)	**88**	78	32	78	78	46	72	49
SVDD(*error*)	**88**	85	**72**	**89**	88	50	82	50
SVM(*no error*)	50	50	53	50	50	43	82	**60**
SVM(*error*)	50	**93**	62	89	**93**	**52**	**90**	46

Table 1: AUCROC Score for SVDD (soft and hard margin) and SVM (soft and hard margin) on anomaly dataset. $C = 0.01$, gamma $\gamma = 1$, $d = 5$

Accuracy | **F1 Score** (%)								
kernels	linear	rbf	poly	sigmoid	laplace	rbfpoly	linrbf	eta kernel
SVDD(*no error*)	**98** |**99**	59 | 72	15 | 23	59|100	59 | 72	21 | 31	57 | 70	17 | 24
SVDD(*error*)	98 | 99	73 | 84	**98 | 99**	84 | 99	81 | 88	**95|97**	77 | 86	**95|97**
SVM(*no error*)	5 | 0	5 | 0	51 | 66	**95|95**	5 | 0	65 | 78	84 | 91	46 | 60
SVM(*error*)	**95 | 97**	**96|98**	76 | 86	87 | 99	**97|98**	16 | 21	**95|91**	9 | 10

Table 2: Accuracy/F1-score for SVDD (soft and hard margin) and SVM (soft and hard margin) on anomaly dataset. $C = 0.01$, gamma $\gamma = 1$, $d = 5$

AUCROC Score (%)								
kernels	linear	rbf	poly	sigmoid	laplace	rbfpoly	linrbf	eta kernel
SVDD(*no error*)	**88**	76	6	69	77	52	74	54
SVDD(*error*)	88	79	**50**	**70**	80	50	77	50
SVM(*no error*)	50	89	**50**	46	76	**69**	76	**63**
SVM(*error*)	51	**92**	11	55	**92**	49	**90**	52

Table 3: AUCROC Score for SVDD (soft and hard margin) and SVM (soft and hard margin) on anomaly dataset. $C = 0.1$, gamma $\gamma = 5$, $d = 10$

Accuracy | **F1 Score** (%)								
kernels	linear	rbf	poly	sigmoid	laplace	rbfpoly	linrbf	eta kernel
SVDD(*no error*)	**98 | 99**	56 | 70	6 | 10	42 | 56	57 | 70	22 | 32	59 | 73	10 | 21
SVDD(*error*)	**98 | 99**	60 | 73	**95 | 97**	43 | 57	57 | 70	**95 | 95**	66 | 78	**95 | 97**
SVM(*no error*)	5 | 0	85 | 91	**95 | 97**	**48 | 63**	54 | 68	51 | 51	58 | 72	34 | 47
SVM(*error*)	51|66	**94|97**	15 | 26	47 | 62	**94|97**	12 | 12	**92 | 95**	10 | 11

Table 4: Accuracy/F1-score for SVDD (soft and hard margin) and SVM (soft and hard margin) on anomaly dataset. $C = 0.1$, gamma $\gamma = 5$, $d = 10$

4 CONCLUSION

In this paper we demonstrated experimentally the use of Support Vector Data Description (SVDD) for Anomaly detection. We concluded that SVDD is best for classifying one class data similar to our anomaly dataset. With a focus on positive class we conclude that SVDD are best performing algorithm for detecting anomaly in one class data than Support Vector Machines (SVMs). We experimented using single and multi-kernels and show that as the value of C increases, the performance of the SVDD decreases- evident from the AUCROC scores.

SVDD is a powerful one class classification algorithm and its linear version performs efficiently in comparison with its kernel version for anomaly detection. We also conclude that Soft margin SVDD kernels can outperform soft-margin SVM kernels in some cases while SVM kernel perform better in other cases.

References

[1] David MJ Tax and Robert PW Duin. Support vector data description. Machine learning, 54(1):45–66, 2004.

[2] V. Chandola, A. Banerjee, and V. Kumar, Anomaly detection: A survey. *ACM computing surveys (CSUR)*,vol. 41, no. 3, p. 15, 2009.

[3] Akoglu L., Tong H., and Koutra D. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[4] Sindagi V. A. and S. Srivastava, Domain adaptation for automatic oled panel defect detection using adaptive support vector data description. *International Journal of Computer Vision*, vol. 122, no. 2, pp. 193–211, 2017.

[5] C. You, D. P. Robinson, and R. Vidal. Provable self representation based outlier detection in a union of subspaces. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*,pp. 1–10., 2017

[6] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, Adversarially learned one-class classifier for novelty detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3379–3388, 2018.

[7] Abati D., Porrello A., Calderara S., and Cucchiara R. AND: Autoregressive novelty detectors. arXiv preprint arXiv:1807.01653, 2018.

[8] Pimentel M.A., Clifton D.A., Clifton L., Tarassenko L. A review of novelty detection. *Signal Processing*, vol. 99, pp. 215–249, 2014.

[9] Markou M., and Singh S. Novelty detection: a reviewpart 1: statistical approaches. *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[10] Zheng S. Smoothly approximated support vector domain description. *Pattern Recognition*, vol 49, pp. 55-64, 2016.

[11] Cortes C., Vapnik V.N. Support Vector Networks. *Machine Learning*, vol 20, no. 3, pp. 273-297, 1995.

[12] Vapnick V. Statistical learning theory. John Wiley. Newyork, 1998.

[13] John Shawt-Taylor, Nello Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, ISBN:9780511809682, pg47-83, 2011.

[14] Mehmet Gönen, Ethem Alpaydin. Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research*, 12:2211-2268, 2011.