Journal of Data Science 23 (2), 287–311
April 2025
DOI: 10.6339/25-JDS1185
Statistical Data Science

Rescale Hinge Loss Support Vector Data Description

Edgard M. Maboudou-Tchao[1], Emil Agbemade[1,*], and Jongik Chung[1]

[1]*Department of Statistics and Data Science, University of Central Florida, Orlando, FL, U.S.A*

Abstract

Significant attention has been drawn to support vector data description (SVDD) due to its exceptional performance in one-class classification and novelty detection tasks. Nevertheless, all slack variables are assigned the same weight during the modeling process. This can lead to a decline in learning performance if the training data contains erroneous observations or outliers. In this study, an extended SVDD model, Rescale Hinge Loss Support Vector Data Description (RSVDD) is introduced to strengthen the resistance of the SVDD to anomalies. This is achieved by redefining the initial optimization problem of SVDD using a hinge loss function that has been rescaled. As this loss function can increase the significance of samples that are more likely to represent the target class while decreasing the impact of samples that are more likely to represent anomalies, it can be considered one of the variants of weighted SVDD. To efficiently address the optimization challenge associated with the proposed model, the half-quadratic optimization method was utilized to generate a dynamic optimization algorithm. Experimental findings on a synthetic and breast cancer data set are presented to illustrate the new proposed method's performance superiority over the already existing methods for the settings considered.

Keywords *anomaly detection; correntropy loss function; hinge loss function; robust one-class classification; rescaled hinge loss function*

1 Introduction

Traditional binary or multi-class classification algorithms sometimes fail to work in real-world datasets because there are only labels for one class, and either no examples for the other class or not enough samples for them. The large quantity of unlabelled data makes this situation difficult because it makes traditional classifiers take longer to train. An approach to this problem is one-class classification (OCC), which seeks to differentiate between normal, lawful transactions and fraudulent, aberrant ones. OCC functions learn from instances that belong to a single class, necessitating complex procedures to attain accurate results. In the larger context of classification tasks, OCC is unique in that it can adjust to imbalanced or limited data availability, aiming for effectiveness even in cases where data from the other class is scarce or nonexistent.

We can better understand the practical use of OCC by examining a few examples from the actual world. The issue of credit cards by banks is a good example to start with. Here, determining whether to grant or deny credit to consumers based on their past financial actions is the main objective. The challenge lies in the rarity of default cases, as most clients pay their bills on time, leading to highly imbalanced datasets. Similarly, in industrial health monitoring—such as for offshore rigs or turbines—data reflecting normal operations is abundant, while instances

*Corresponding author. Email: agbemil@gmail.com.

Received August 17, 2024; Accepted April 7, 2025

of anomalies or system failures are far less common. These failures, however, are critical for safety and preventative maintenance. For example, an undetected malfunction in an aircraft's engine vibration sensor or landing gear hydraulic system could lead to serious safety risks. This scarcity of anomalies highlights the importance of OCC, which leverages the dominant class to detect outliers, even in the absence of many failure examples. A primary goal of many OCC methods is to define a decision boundary that effectively encloses the target class within the training dataset. Establishing robust decision limits is essential, as OCC aims to identify hidden outliers while preserving the integrity of the target class.

Based on the model of the classifier, the data type, and the temporal dynamics of the features, Khan and Madden (2014) categorize OCC approaches. They distinguish between three main types of OCC models: those that use density to determine the target class's data distribution, those that use boundaries to encapsulate the data within a certain area, and those that use reconstruction to re-create data points and identify outliers by looking at reconstruction errors.

Two key parameters determine whether a data point is an outlier or an inlier in the OCC framework: one evaluates the distance of a data point from the target class, and the other is a user-defined threshold that decides whether the data point is accepted or rejected as an inlier (Kennedy et al., 2009).

Density-based one-class classification methods work by calculating the training data's density and comparing it to a predetermined threshold—a model parameter. These techniques work well with well-populated datasets that have a significant number of training examples. Gaussian method, a mixture of Gaussians, and Parzen density estimation are a few density-based methods (Seliya et al., 2021). Boundary-based approaches, in contrast, concentrate on drawing a clear boundary around the data points that are regarded as the target class. To do this, a closed border must be drawn, and any data point found outside of it is considered an outlier. The main difficulty with these approaches is optimizing this boundary for precise modeling. This strategy is demonstrated by the One-class Support Vector Machine (OCSVM), a kernel-based technique based on Support Vector Machines (SVMs) that creates a hyperplane by maximizing the distance from the origin, which distinguishes the target class from outliers (Schölkopf et al., 1999). Similar to this, the target samples are encircled by a minimal radius hypersphere formed by the Support Vector Data Description (SVDD) approach; outliers are those samples that fall outside of the sphere (Tax and Duin, 2004). Boundary-based approaches can perform comparably with fewer data samples than density-based approaches.

Sun and Tsung (2003) proposed to use SVDD to construct control charts to detect changes in a process. Similarly, Maboudou-Tchao (2021b) suggested control charts based on Least Squares support vector data description (LS-SVDD) to detect outliers. Maboudou-Tchao et al. (2018) proposed to use Mahalanobis kernels and rational subgroups with SVDD for outlier detection. Ruff et al. (2018) proposed a deep neural network and SVDD for one-class classification. Maboudou-Tchao and Hampton (2025) suggested a deep neural network and LS-SVDD for anomaly detection. Maboudou-Tchao and Harrison (2021) compare SVDD and ℓ_2-norm SVDD in a variety of settings, propose an SMO algorithm for SVDD using an ℓ_2 norm, and provide algorithms to compute the solutions for the unconstrained SVDD and ℓ_2-SVDD primal problems. Maboudou-Tchao (2020) proposed a control chart based on the Least Squares OCSVM, which is a least-squares reformulation of OCSVM. For higher order problems, Maboudou-Tchao (2018) proposed Support Matrix Data Description (SMDD) for one-class classification in matrices with an application to covariance matrices. Maboudou-Tchao (2021c, 2023) proposed Support Tensor Data Description and Least Squares Support Tensor Data Description, respectively, for one-class classification in tensor-variate data with applications to image datasets. Maboudou-Tchao

(2021a) proposed a one-class classification method based on Support Tensor Vector Data Description to detect outliers with high dimensional vectors.

Our modeling approach is designed to accommodate varying dimensional settings rather than being restricted to specific feature counts. To illustrate this flexibility, we conducted experiments with lower-dimensional settings ($p = 5$, $p = 10$) and a higher-dimensional scenario ($p = 100$). These choices were made to demonstrate the model's adaptability across different feature spaces. However, the framework is not limited to these dimensions; users can train the model with any available feature set, making it applicable to a wide range of real-world datasets and tasks.

In our study, we introduce an innovative rescale hinge loss SVDD model that redefines the conventional optimization challenge of SVDD by integrating a rescaled hinge loss function. This advancement led to the development of a dynamic optimization algorithm tailored to enhance the model's functionality. We further explore the relationship between our novel approach and existing weighted SVDD techniques, highlighting its superior capability in executing OCC tasks, particularly in scenarios where the data are compromised by outliers. To validate the efficiency and performance of our proposed method, we focused on comparing our methodology with the Support Vector Data Description (SVDD), Density Weighted SVDD (DW-SVDD), Stahel–Donoho SVDD (SD-SVDD) across numerous OCC tasks. This focused approach allowed us to deeply analyze and demonstrate our model's performance and utility in real-world applications.

2 Related Works

2.1 Support Vector Data Description (SVDD)

Support Vector Data Description (SVDD) is introduced by Tax and Duin (2004). This model aims to encapsulate the data points representative of the primary class within a hypersphere while excluding all extraneous points. Consider a set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ of independent observations. They approach the optimization problem of SVDD with the following constrained loss function formulation:

Minimize the objective function as formulated by Boyd and Vandenberghe (2004):

$$\min_{R,\mu,\xi} R^2 + C\sum_{i=1}^{N}\xi_i, \tag{1}$$

subject to the constraints:

$$\|\phi(\mathbf{x}_i) - \mu\|^2 \leqslant R^2 + \xi_i, \quad \xi_i \geqslant 0, \quad i = 1, 2, \ldots, N, \tag{2}$$

where μ represents the hypersphere's centroid or center and R represents the radius, C acts as a penalty coefficient, and $\xi = \{\xi_i\}_{i=1}^N$ represents the error slack. The transformation $\phi(\mathbf{x}_i)$ corresponds to the high-dimensional feature space projection, known to be a reproducing kernel Hilbert space (RKHS) (Kivinen et al., 2004). We encourage readers who are not familiar with convex optimization problems to refer to Boyd and Vandenberghe (2004) for a better understanding of some of the terminologies (i.e., error slack, slack variable).

The kernel inner product $\langle\phi(\mathbf{x}_1), \phi(\mathbf{x}_2)\rangle$ is computable via a kernel function $k(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_1)^\top\phi(\mathbf{x}_2)$. Notably, with a Gaussian kernel, we define $k(\mathbf{x}_1, \mathbf{x}_2)$ as:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}\right). \tag{3}$$

Equations (1) and (2) can be rewritten as an unconstrained loss optimization problem as:

$$\min_{R,\mu} \quad R^2 + C\sum_{j=1}^{N} \max\left\{0, \left\|\phi(\mathbf{x}_i) - \mu\right\|^2 - R^2\right\}.$$

However, the dual optimization problem employs the Lagrange multiplier technique (Tax and Duin, 2004), transforming the primary problem into:

Minimize:

$$\min_{\alpha} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}_i), \tag{4}$$

subject to:

$$\sum_{i=1}^{N} \alpha_i = 1, \quad 0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \ldots, N. \tag{5}$$

Here, $\alpha_1, \ldots, \alpha_N$ are the Lagrange multipliers, collected in the vector α. The revised optimization scheme, represented in equation (4), maintains adherence to the slack variable constraints and ensures the sum of α equals unity, a condition for optimization in the RKHS framework.

Upon resolving the dual problem using established quadratic programming methodologies, the hypersphere's parameters μ and R are derivable from α using the following equations:

The center μ is given by:

$$\mu = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i), \tag{6}$$

and the radius R is:

$$R = \|\phi(\mathbf{x}_*) - \mu\|, \tag{7}$$

which can be expanded based on the kernel function as:

$$R = \sqrt{k(\mathbf{x}_*, \mathbf{x}_*) - 2\sum_{i=1}^{N} \alpha_i k(\mathbf{x}_*, \mathbf{x}_i) + \beta}, \tag{8}$$

where $\beta = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$ is the sum of the products of the Lagrange multipliers for all support vectors, and $\mathbf{x}_*$ is a support vector within the set of $\{\mathbf{x}_i \mid 0 < \alpha_i < C,\ i = 1, 2, \ldots, N\}$.

Upon receiving a new test input $\mathbf{x}$, its decision function $d(\mathbf{x})$ is:

$$d(\mathbf{x}) = \left\|\phi(\mathbf{x}) - \mu\right\|^2 - R^2. \tag{9}$$

The decision function evaluates the distance from the center in the feature space and compares it to R^2.

This is further expressed as:

$$d(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - 2\sum_{i=1}^{N} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + \beta - R^2. \tag{10}$$

A negative value of $d(\mathbf{x})$ implies that $\mathbf{x}$ belongs to the target class (inside the hypersphere), otherwise, it is categorized as a non-target class sample (outside the hypersphere).

2.2 Review of Some Robust Methods

Support vector data description (SVDD) is a widely used tool for the one-class classification problem (Tax and Duin, 1999, 2004). However, it is heavily affected by the presence of an even very small fraction of contamination, caused by errors in the measurement of feature values or mislabeling. In this case, the trained classifier will sometimes tend to enclose objects which are remote from the target class. When the training data contain noise or uncertainty, the noise data may behave like normal, and be enclosed inside the hyper-sphere in the training processes. Consequently, the spherical boundary may not be optimal and the detection performance will become bad. There is a need to find a more reliable and compact description of the target set. Many approaches have been proposed in the literature. We will quickly review two of them below.

2.2.1 Weighted SVDDs

Assigning variable weights to slack variables introduces a category of SVDD approaches known as weighted SVDDs, as documented in the works of Wang and Lai (2013); Cha et al. (2014); Wang and Lan (2020); Hu et al. (2021). These methodologies have been validated for their efficiency in bolstering the robustness of the SVDD framework. Although these methods employ diverse strategies for weight calculation, their optimization problems can be encapsulated in a singular representation:

$$\min_{R,\mu,\xi} R^2 + C\sum_{i=1}^{N} \mathbf{w}_i \xi_i, \tag{11}$$

subject to the constraints:

$$\left\|\phi(\mathbf{x}_i) - \mu\right\|^2 \leqslant R^2 + \xi_i, \quad \xi_i \geqslant 0, \quad i = 1, \ldots, N. \tag{12}$$

Here, $\{\mathbf{w}_i\}_{i=1}^N$ represent a set of predetermined weights. Taking the density-weighted approach by Cha et al. (2014) as an example, weights are determined as follows:

$$\mathbf{w}_i = 1 - \frac{d(\mathbf{x}_i, \mathbf{x}_i^{(k)})}{\max\{d(\mathbf{x}_1, \mathbf{x}_1^{(k)}), \ldots, d(\mathbf{x}_N, \mathbf{x}_N^{(k)})\}}. \tag{13}$$

In this equation, $\mathbf{x}_i^{(k)}$ symbolizes the k-th nearest neighbor of $\mathbf{x}_i$, and $d(\mathbf{x}_i, \mathbf{x}_i^{(k)})$ measures the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_i^{(k)}$. The resulting distances and hence the weights—tend to be smaller for outliers in comparison to target points because outliers usually lie in sparser locations. Following the debate in the literature by Wang and Lai (2013), Wang and Lan (2020), and Hu et al. (2021), this weighting system emphasizes the contribution of the members of the target class while minimizing the impact of outliers. After pre-calculating $\{\mathbf{w}_i\}_{i=1}^N$, the dual optimization problem is approached via the Lagrange multiplier method akin to standard SVDD, formalized as follows:

$$\min_{\alpha} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}_i), \tag{14}$$

subject to:

$$\sum_{i=1}^{N} \alpha_i = 1, \tag{15}$$

$$0 \leqslant \alpha_i \leqslant \mathbf{w}_i C, \quad i = 1, 2, \ldots, N. \tag{16}$$

From the formulation, it is evident that each Lagrange multiplier is bounded above by a product of its corresponding weight and the regularization parameter C, symbolized by $\alpha_i \leqslant \mathbf{w}_i C$. This constraint facilitates a more refined equilibrium between the data representing the class of interest and the anomalous data points, ensuring that the weights when judiciously determined, enhance the overall classification accuracy. Here, to determine if a new point is in a target class or an outlier, the decision function used in (9) and (10) above was utilized.

2.2.2 Stahel–Donoho SVDD

To enhance robustness against contamination, such as outliers or mislabeled data, the proposed SD-SVDD (Wang and Lan, 2020) introduces weights into the SVDD framework based on the Stahel–Donoho (SD) outlyingness computed in a kernel-induced feature space (Stahel, 1981; Donoho, 1982). The key idea is to assign smaller weights to observations that exhibit higher outlyingness, thereby reducing their influence on the boundary of the hypersphere that defines the target class.

The weight function $w(\tilde{r}_i)$, where $\tilde{r}_i$ is the kernel SD outlyingness of point $\mathbf{x}_i$, is designed to decrease smoothly from 1 to 0 as outlyingness increases. Two types of weight functions are considered. The first is the hard rejection rule defined as

$$w_{\mathrm{hr}}(\tilde{r}_i) = \mathbb{I}(\tilde{r}_i \leqslant c),$$

where $\mathbb{I}(\cdot)$ is the indicator function. This function assigns a weight of 1 to observations with outlyingness less than or equal to a threshold c, and 0 otherwise, effectively removing highly outlying observations. The second is the Huber-type weight function which softens the rejection by assigning decreasing weights to more outlying samples:

$$w_H(\tilde{r}_i) = \mathbb{I}(\tilde{r}_i \leqslant c) + \left(\frac{c}{\tilde{r}_i}\right)^q \mathbb{I}(\tilde{r}_i > c),$$

where $c = \mathrm{median}(\tilde{r}) + \mathrm{mad}(\tilde{r})$ and $q = 3$ is a shape parameter. This form offers a trade-off between robustness and efficiency, making it more adaptable to varying degrees of contamination in the data. This objective aims to find a minimum volume hypersphere in the kernel space that encloses most of the weighted data. The optimization problem is expressed as

$$\min_{R,\xi_i,\mu} R^2 + C\sum_{i=1}^{n} w_i\xi_i,$$

subject to

$$\left\|\phi(\mathbf{x}_i) - \mu\right\|^2 \leqslant R^2 + \xi_i, \quad \xi_i \geqslant 0, \quad i = 1, \ldots, n.$$

Here, $\phi(\cdot)$ denotes the mapping to the kernel feature space, ξ_i are slack variables allowing some target samples to lie outside the sphere, and C is the regularization parameter balancing the trade-off between model complexity and misclassification.

To solve this, the dual form is derived by introducing Lagrange multipliers α_i, resulting in the following optimization:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j),$$

subject to

$$\sum_{i=1}^{n} \alpha_i = 1, \quad 0 \leqslant \alpha_i \leqslant w_i C.$$

The center of the hypersphere in feature space is then given by

$$\mathbf{a} = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i),$$

and the decision function for evaluating a new test point $\mathbf{x}$ is computed as

$$f(\mathbf{x}) = \kappa(\mathbf{x}, \mathbf{x}) - 2\sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}, \mathbf{x}_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) - R^2.$$

A test point is accepted as part of the target class if $f(\mathbf{x}) \leqslant 0$, and rejected as an outlier otherwise. This formulation allows each training sample to contribute differently to the decision boundary, depending on its degree of outlyingness.

2.3 Comparative Summary and Limitations of Reviewed Methods

Despite their shared foundation in the SVDD framework, the methods presented differ in their strategies for addressing noise and outliers, each with distinct advantages and drawbacks. SVDD assumes that the training data is clean and representative of the target class, making it highly sensitive to even minor contamination. This weakness often results in overestimating the hypersphere radius and poor generalization to new data. Weighted SVDD variants attempt to improve robustness by adjusting slack penalties based on distance or density estimates. However, these methods depend heavily on accurate weight estimation, which may be unreliable in high-dimensional or highly clustered data. Furthermore, inappropriate parameter choices—such as the number of nearest neighbors in density-based weighting—can misclassify target instances as outliers. Stahel–Donoho SVDD (SD-SVDD) leverages kernel-based outlyingness to assign adaptive weights, significantly improving robustness against severe contamination. However, calculating SD outlyingness involves computationally intensive procedures such as projection pursuit in high-dimensional kernel spaces, which can limit scalability. Moreover, the choice between hard rejection and Huber-type weighting introduces an extra layer of hyperparameter tuning and trade-offs between robustness and efficiency.

3 Methodology

The introduction of SVDD by Tax and Duin (2004) has been a landmark achievement in the realm of machine learning. Yet, despite its effectiveness, the SVDD framework exhibited vulnerabilities, particularly its sensitivity to anomalies. These anomalies could skew the boundary defined by the decision functions, leading to less accurate identification of outliers.

Recognizing these limitations, the scientific community sought to evolve the SVDD framework to enhance its robustness (Tax and Duin, 2004, 1999; Ghasemi et al., 2021; Erfani et al., 2016; Yang et al., 2017). This pursuit resulted in the development of Rescale Hinge Loss Support Vector Data Description (RSVDD). RSVDD aims to fortify the model against the influence of

anomalies, ensuring the integrity of the descriptive boundary amidst noisy or outlier-rich environments. The fundamental strategy for achieving this robustness involves significantly altering the SVDD's optimization problem.

The enhancement process begins with introducing a rescaled hinge loss function, which modifies the penalty for data points outside the descriptive boundary. Unlike the traditional hinge loss function, which increases linearly and is unbounded, the rescaled version introduces a bounded, non-linear penalty. This adjustment allows the model to maintain flexibility and sensitivity to the dataset's structure while mitigating the undue influence of outliers.

Moreover, to navigate the complexities introduced by this new loss function, the study uses the Half-Quadratic (HQ) optimization (Wright et al., 2008). HQ optimization is adept at handling the challenges posed by non-convex optimization problems, which are common when dealing with non-linear and bounded loss functions. By employing this method, the study presents a dynamic optimization algorithm specifically designed for the RSVDD model. This algorithm iteratively refines the model parameters, effectively balancing between fitting the majority of data points and ignoring anomalies. Through these methodological advancements, the RSVDD model emerges as a more resilient tool in the machine learning arsenal, capable of providing accurate and reliable outlier detection across a wide array of applications.

3.1 Adaptive Rescaling of Hinge Loss

Correntropy represents a metric derived from information theory, serving as a cornerstone in robust machine learning paradigms. It has been applied in robust learning algorithms, such as sparse representation classifiers. It forms the foundation for Correntropy-induced loss functions that have shown promising results in robust face recognition and neural network training (Liu et al., 2007; Principe, 2010; Wright et al., 2008).

In robust machine learning, the hinge loss function is commonly used for classification tasks, especially within the Support Vector Machine framework. However, this function is unbounded, which may lead to a sensitivity to outliers. Singh et al. (2014) suggested using the Correntropy idea in classification task and proposed the Correntropy loss function or C-loss:

$$\ell_c(z) = \beta\left[1 - \exp\left(-\frac{(1-z)^2}{2\sigma^2}\right)\right], \tag{17}$$

where σ denotes the bandwidth parameter, and $\beta = [1 - \exp(-\frac{1}{(2\sigma^2)})]^{-1}$ is a normalization constant ensuring $\ell_c(0) = 1$.

We formulate our rescaled hinge loss function following the idea of the C-loss as:

$$\ell_{rhinge}(z_j) = \beta\big[1 - \exp\big(-\eta\ell_{\text{hinge}}(z_j)\big)\big], \tag{18}$$

where $\ell_{\text{hinge}}(z_j) = \max\{0, \|\phi(\mathbf{x}_j) - \mu\|^2 - R^2\}$ and $\eta = \frac{1}{(2\sigma^2)} > 0$. This formulation ensures the loss remains monotonic, bounded, non-convex, and smooth, adapting to the underlying data distribution effectively. Consequently, using a bounded version of the hinge loss function instead of the unbounded hinge loss function used in the SVDD will help us solve the optimization problem of RSVDD.

Proposition 1. $\lim_{\eta\to 0} \ell_{rhinge}(z_j) = \ell_{\text{hinge}}(z_j)$.

The proof of this proposition can be found in Xu et al. (2017).

In the subsequent section, we integrate this rescaled hinge loss into the SVDD formulation to enhance its robustness against outliers and noise.

3.2 RSVDD Based on $\ell_{rhinge}(z_j)$

Let's consider the following unconstrained SVDD optimization problem:

$$\min_{R,\mu} \quad R^2 + C\sum_{j=1}^{N} \max\left\{0, \left\|\phi(\mathbf{x}_j) - \mu\right\|^2 - R^2\right\}, \tag{19}$$

which is equivalent to:

$$\min_{R,\mu} \quad R^2 + C\sum_{j=1}^{N} \ell_{\text{hinge}}(z_j). \tag{20}$$

By replacing the unbounded hinge loss in (19) with the bounded rescaled hinge loss function we can obtain the optimization problem of RSVDD as:

$$\min_{R,\mu} \mathcal{L}_{l1}(R,\mu) = \min_{R,\mu} \quad R^2 + C\sum_{j=1}^{N} \ell_{rhinge(z_j)}, \tag{21}$$

where $\ell_{rhinge(z_j)} = \beta[1 - \exp\{-\eta l_{\text{hinge}}(z_j)\}]$, $C \geqslant 0$, $\beta = \frac{1}{1-\exp\{-\eta\}}$ and $\mathcal{L}_{l1}(R,\mu) = R^2 + C\sum_{j=1}^{N} \ell_{rhinge(z_j)}$.

By simple modification, (21) becomes

$$\max_{R,\mu} \mathcal{L}_{l2}(R,\mu) = \max_{R,\mu} \quad -R^2 + C\beta\sum_{j=1}^{N}\left[\exp\left\{-\eta l_{\text{hinge}}(z_j)\right\}\right], \tag{22}$$

where $\mathcal{L}_{l2}(R,\mu) = -\mathcal{L}_{l1}(R,\mu)$.

We recognize that the above-rescaled hinge loss function is non-convex and can be solved using the idea of Half-Quadratic (HQ) optimization technique (Nikolova and Ng, 2005) for non-convex functions.

We derived that (22) is equivalent to

$$\max_{R,\mu,\mathbf{u}} \mathcal{L}_{l3}(R,\mu,\mathbf{u}) = \max_{R,\mu,\mathbf{u}} \quad -R^2 + C\beta\sum_{j=1}^{N}\left\{\eta l_{\text{hinge}}(z_j)u_j - g(u_j)\right\}. \tag{23}$$

Details of the derivation of (23) can be found in Appendix B.

Now, we can solve (23) using the alternating optimization approach. Specifically, given (R^τ, μ^τ), we optimize over $\mathbf{u}^\tau$ and with a fix $\mathbf{u}^\tau$, we can get $(R^{\tau+1}, \mu^{\tau+1})$ together with the τ here denoting τth iteration, the optimization problem (23) is equivalent to

$$\max_{u_j<0} \quad \sum_{j=1}^{N}\left\{-\eta l_{\text{hinge}}(z_j)u_j - g(u_j)\right\}, \tag{24}$$

where $g(u_j)$ represents a convex function introduced to manage the re-scaled hinge loss function's non-convexity.

However, we know that (24) has an analytical solution:

$$u_j^\tau = -\exp\left\{-\eta l_{\text{hinge}}\left(z_j^\tau\right)\right\}, \quad j = 1, 2, \ldots, N. \tag{25}$$

Now, with a fix u_j^τ in (25), we can optimize $(R^{\tau+1}, \mu^{\tau+1})$ by solving the following:

$$\min_{R,\mu} \quad R^2 + C\sum_{j=1}^{N} s_j l_{\text{hinge}}(z_j), \tag{26}$$

where $s_j = -\beta\eta u_j > 0$.

The weights (s_j) are derived from the re-scaled hinge loss function and act as an adaptive regularization mechanism. Unlike density-based weights (e.g., in Weighted SVDD), our weights prioritize well-classified points by assigning them higher influence while reducing the impact of harder-to-classify points. This ensures a more stable optimization process and helps refine the decision boundary effectively.

The dual optimization of (26) is derived as:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{j=1}^{N} \alpha_j k(\mathbf{x}_j, \mathbf{x}_j) - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{j=1}^{N} \alpha_j = 1, \\ & 0 \leqslant \alpha_j \leqslant C_j, \quad 1 \leqslant j \leqslant N, \end{aligned} \tag{27}$$

where $C_j = Cs_j = -C\beta\eta u_j$.

After solving (33), the decision rule becomes

$$k(\mathbf{z}, \mathbf{z}) - 2\sum_{j}^{N} \alpha_j k(\mathbf{x}_j, \mathbf{z}) + \sum_{i,j}^{N} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \leqslant R^2. \tag{28}$$

An unseen data, $\mathbf{z}$, is a target if the above condition is true. Otherwise, $\mathbf{z}$ is classified as an outlier.

The squared radius is computed as:

$$R^2 = \frac{1}{N_s}\sum_{s=1}^{N_s}\Bigg(k(\mathbf{x}_s, \mathbf{x}_s) - 2\sum_{i=1}^{N_s} \alpha_i k(\mathbf{x}_s, \mathbf{x}_i) \quad + \sum_{i,j=1}^{N_s} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)\Bigg), \tag{29}$$

where N_s is the total number of the support vectors and $\mathbf{x}_s$ are the support vectors.

We employ Algorithm 1 to optimize the objective function in Equation (28) using the HQ optimization method. The training dataset, denoted as $\{\mathbf{x}_i\}_{i=1}^{N}$, serves as input to this algorithm, where it iteratively refines the model parameters R and $\boldsymbol{\mu}$. During each iteration, auxiliary variables $\mathbf{u}$ are updated based on the current model parameters, ensuring a progressive improvement in the optimization process. The algorithm runs until convergence or until a predefined maximum number of iterations $T_{\max}$ is reached, at which point it returns the optimized parameters that define the decision boundary of the model. Once the training phase is completed, Algorithm 2 is used to determine a threshold h, which is essential for classifying new observations as either belonging to the target class or as outliers. The training dataset is used in this step to compute the distances of the training points from the learned decision boundary. A bootstrap-based approach is then applied to estimate the threshold h, ensuring that it aligns with the desired Type I error rate. The computed threshold enables a more controlled and statistically reliable classification process when applying the model to unseen data. Together, Algorithm 1 learns the model parameters using the training dataset, while Algorithm 2 establishes a data-driven threshold for decision-making, enhancing the robustness of the anomaly detection framework.

We use the HQ optimization method for equation (22), focusing mainly on steps (25), (27) and (29). The full process is shown in Algorithm 1.

Proposition 2. *The sequence* $\{\mathcal{L}_{l3}(R^\tau, \mu^\tau, \mathbf{u}^\tau), \tau = 1, 2, \dots\}$ *produced by Algorithm 1 converges.*

Appendix C contains the proof for Proposition 2.

Algorithm 1 HQ optimization algorithm for (22).

Input: Training set $\{\mathbf{x}_i\}_{i=1}^N$, trade-off parameter v, scale constant η in $l_{rhinge(z_j)}$;the regularization parameter C; the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$; maximum number of iterations $T_{\max}$.
Output: R, $\boldsymbol{\mu}$ in (27) and (29).
Initialization: Number of iterations $\tau = 0$, vector of auxiliary variables $\mathbf{u} = [-1, -1, \ldots, -1]^T \in \mathbb{R}^N$.
while $\tau < T_{\max}$ **do**
    Obtain $R^{\tau+1}$ and $\boldsymbol{\mu}^{\tau+1}$ by solving (27).
    Update $\mathbf{u}^{\tau+1}$ by (25).
    Set $\tau = \tau + 1$.
end while
return $R = R^{\tau+1}$ and $\boldsymbol{\mu} = \boldsymbol{\mu}^{\tau+1}$

3.3 Selection of the Threshold (h)

To determine a threshold for detecting outliers or classifying observations, one can use the radius R^2. The radius R^2 can help identify outliers by comparing the distance of a new observation to the target class. However, it doesn't allow for controlling Type I error rates, which is the likelihood of incorrectly identifying an observation as an outlier. To manage Type I errors, a threshold h is determined based on the desired error rate α, ensuring that an observation is classified as a target if its distance d is less than or equal to h. The threshold h is usually calculated through a bootstrap simulation technique, which helps achieve the specified Type I error rate. Summary of determining the threshold is shown in Algorithm 2.

Algorithm 2 Bootstrap algorithm for threshold h.

Input: Training dataset $\{\mathbf{x}_i\}_{i=1}^N \subseteq \mathbb{R}^p$ with $x_i \in \mathbb{R}^p$, B bootstrap samples, and hypersphere center μ^* of a trained model.
Output: A threshold h
for $x_i \in D$ **do**
    $d_i \leftarrow k(\mathbf{x}, \mathbf{x}) - 2\sum_j^N \alpha_j k(\mathbf{x}_j, \mathbf{x}) + \sum_{i,j}^N \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$
end for
for $b = 1$ **to** B **do**
    Draw a bootstrap sample of size N from the set of N d statistics
    If α is the desired Type I error, determine the $100 \times (1 - \alpha)$ percentile value
end for
Obtain the threshold h by taking an average of $B \times 100 \times (1 - \alpha)$ percentile values

3.4 Computational Complexity Analysis

It is widely recognized that solving the dual optimization problem integral to the One-Class Support Vector Machines (OCSVM) requires a computational effort of $O(N^3)$ (Schölkopf et al., 2001; Khan et al., 2014), where N denotes the number of training samples. Consequently, the complexity for calculating the Lagrange multiplier vector per iteration within a RSVDD framework adheres to the same $O(N^3)$ complexity. Additionally, the computational requirements for updating the auxiliary vector $\mathbf{u}$ in each iteration is $O(N^2)$.

Given these complexities, the aggregate computational burden of the complete algorithm can be succinctly described as $O(I_{\mathrm{HQ}}(N^3 + N^2))$, where I_{HQ} is the count of half-quadratic optimization iterations. Disregarding lesser order terms, this collectively simplifies the computational complexity of Algorithm 1 to $O(I_{\mathrm{HQ}}N^3)$, representing a significant computational demand as the number of training samples increases.

4 Performance Study

4.1 Uncorrelated Multivariate Normal Data

4.1.1 Training Data Generation and Structure

The training set consists of N independent random vectors drawn from a mixture of two different multivariate normal distributions. Specifically, the training data are divided into two groups:

- The first sample consists of n_1 vectors drawn from a multivariate normal distribution with mean vector $\boldsymbol{\mu}_1 = \mathbf{0}$ and an identity covariance matrix $\boldsymbol{\Sigma} = \mathbf{I}$.
- The second sample consists of n_2 vectors drawn from a multivariate normal distribution with a shifted mean vector $\boldsymbol{\mu}_2$, where the first component is γ and the remaining components are zero while retaining the same identity covariance matrix.

First Sample (n_1 Vectors): Each vector $\mathbf{x}_i$ in the first sample follows:

$$\mathbf{x}_i \sim \mathcal{N}(\mu_1, \boldsymbol{\Sigma}),$$

where

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^p, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{p \times p}.$$

Second Sample (n_2 Vectors): Each vector $\mathbf{x}_i$ in the second sample follows:

$$\mathbf{x}_i \sim \mathcal{N}(\mu_2, \boldsymbol{\Sigma}),$$

where

$$\boldsymbol{\mu}_2 = \begin{bmatrix} \gamma \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^p, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{p \times p}.$$

The parameter γ is chosen based on the dimensionality:

- $\gamma = 0.5$ for $p = 5$,
- $\gamma = 5$ for $p = 10$ and $p = 100$.

Finally, the two samples are shuffled and combined to form the contaminated training set $\mathcal{D}$:

$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}.$$

We conduct simulations for different values of p to assess the model's behavior under varying dimensional settings:

- For $p = 5$ and $p = 10$, we use a training set of $N = 100$, with $n_1 = 60$ and $n_2 = 40$.
- For $p = 100$, the training set size is reduced to $N = 40$, with $n_1 = 25$ and $n_2 = 15$.

4.1.2 Test Set Generation and Evaluation

The test set consists of 15,000 independent vectors, $\mathbf{x}_i$, for $i = 1, 2, \ldots, 15{,}000$, drawn from a multivariate normal distribution with a mean shift parameter δ and an identity covariance matrix $\mathbf{\Sigma}$. This configuration allows us to evaluate the model's performance under different degrees of mean shift. Mathematically, the test vectors are sampled as:

$$\mathbf{x}_i \sim \mathcal{N}(\mu, \mathbf{\Sigma}) \quad \text{for } i = 1, 2, \ldots, 15{,}000,$$

where:

$$\boldsymbol{\mu} = \begin{bmatrix} \delta \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^p, \quad \mathbf{\Sigma} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{p \times p}.$$

The value of δ varies depending on the dimension p, allowing us to assess the model's sensitivity to different levels of shift:

- For $p = 5$: $\delta \in \{0, 0.2, 0.4, \ldots, 1, 1.5, 2, \ldots, 6\}$.
- For $p = 10$: $\delta \in \{0, 1, 2, \ldots, 10\}$.
- For $p = 100$: $\delta \in \{0, 2.5, 5, 6, 7.5, 9, 10.5\}$.

A total of 15,000 test observations are generated for each case, ensuring a comprehensive evaluation of the model's performance. The performance of our proposed method is compared against three benchmark algorithms: Support Vector Data Description (SVDD), Density-Weighted SVDD (DW-SVDD), Stahel–Donoho SVDD (SD-SVDD). The primary evaluation metric is the Type II error probability (false negative rate), measured under a fixed Type I error probability of 0.05. This allows us to analyze the detection power of the models when distinguishing between normal and shifted distributions.

4.2 Correlated Multivariate Normal Data

4.2.1 Training Data Generation and Structure

In this scenario, the training set is generated from a correlated multivariate normal distribution with a structured covariance matrix $\mathbf{R}_0$. The mean vector for the majority of the training data is set to zero, while a subset of the data uses a different mean vector to simulate the contamination. Mathematically, the first sample (n_1) is drawn as:

$$\mathbf{x}_i \sim \mathcal{N}(\mu_0, \mathbf{R}_0),$$

where:

$$\boldsymbol{\mu}_0 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^p, \quad \mathbf{R}_0 = \begin{bmatrix} 2 & 0.7^{|i-j|} & \cdots & 0.7^{|i-j|} \\ 0.7^{|i-j|} & 2 & \cdots & 0.7^{|i-j|} \\ \vdots & \vdots & \ddots & \vdots \\ 0.7^{|i-j|} & 0.7^{|i-j|} & \cdots & 2 \end{bmatrix} \in \mathbb{R}^{p \times p}.$$

The contaminated sample (n_2) follows the same covariance structure but with a shifted mean:

$$\mathbf{x}_i \sim \mathcal{N}(\mu_c, \mathbf{R}_0),$$

where:

$$\boldsymbol{\mu}_c = \begin{bmatrix} \gamma \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^p.$$

Similarly, the parameter γ is used to simulate contamination in the dataset and is chosen based on the dimensionality, with $\gamma = 0.5$ for $p = 5$, and $\gamma = 5$ for $p = 10$. Each diagonal element $\mathbf{R}_0$ is set to 2, ensuring unit variance, while the off-diagonal elements introduce correlation, where $r_{ij} = 0.7^{|i-j|}$ for $i \neq j$, meaning the correlation decays exponentially with distance. We shuffle and combine the two samples to form the contaminated training set. $\mathcal{D}$:

$$\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}.$$

4.2.2 Test Set Generation and Evaluation

The test set consists of 15,000 independent vectors $\mathbf{x}_i$, for $i = 1, 2, \ldots, 15{,}000$, drawn from a correlated multivariate normal distribution with a mean shift parameter δ and a structured covariance matrix $\mathbf{R}_0$, where the value δ varies depending on the dimension p: for $p = 5$, δ ranges from 0 to 6 in increments of 0.2 and 0.5; for $p = 10$, it ranges from 0 to 10 in steps of 1. This configuration allows us to evaluate the model's performance under varying levels of mean shift. Mathematically, the test vectors are sampled as:

$$\mathbf{x}_i \sim \mathcal{N}(\mu, \mathbf{R}_0),$$

where:

$$\boldsymbol{\mu} = \begin{bmatrix} \delta \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^p, \quad \mathbf{R}_0 = \begin{bmatrix} 2 & 0.7^{|i-j|} & \cdots & 0.7^{|i-j|} \\ 0.7^{|i-j|} & 2 & \cdots & 0.7^{|i-j|} \\ \vdots & \vdots & \ddots & \vdots \\ 0.7^{|i-j|} & 0.7^{|i-j|} & \cdots & 2 \end{bmatrix} \in \mathbb{R}^{p \times p}.$$

4.3 Uncorrelated Simulation Results and Discussion

Figures 1, 2, and 3 show the Type II error rates of RSVDD, SVDD, SD-SVDD, and DW-SVDD for varying values of δ under different dimensions: $p = 5$, $p = 10$, and $p = 100$, respectively. In all cases, the error rates across methods start at similar levels when $\delta = 0$, and consistently decrease as δ increases, indicating better anomaly detection with stronger shifts. As dimensionality increases, differences between the methods become more evident. For $p = 5$, the four methods, closely aligned curves are shown, with RSVDD showing consistently lower Type II error rates across all δ values, though the gap is relatively narrow. In the $p = 10$ setting, RSVDD maintains a visible advantage, especially beyond $\delta = 3$, where the separation between RSVDD and the other methods becomes more pronounced. At $p = 100$, the downward trends persist across all models, but RSVDD continues to show a consistently lower Type II error curve, especially in the mid-to-high δ range, distinguishing itself more clearly from the alternatives. Overall, while all models improve with increasing δ, the observed differences across dimensions highlight RSVDD's consistent performance advantage, which becomes more noticeable as the dimensionality grows.

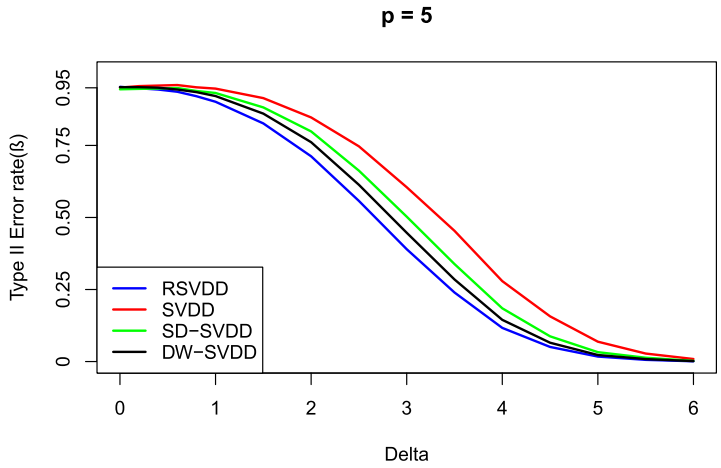

Figure 1: Type II error plot for $p = 5$.

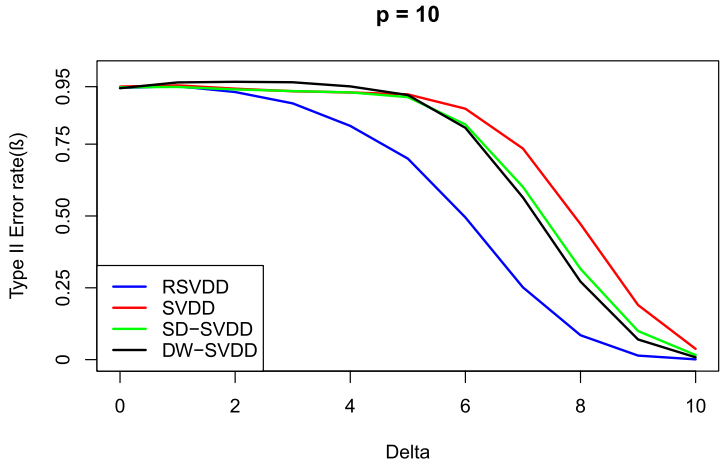

Figure 2: Type II error plot for $p = 10$.

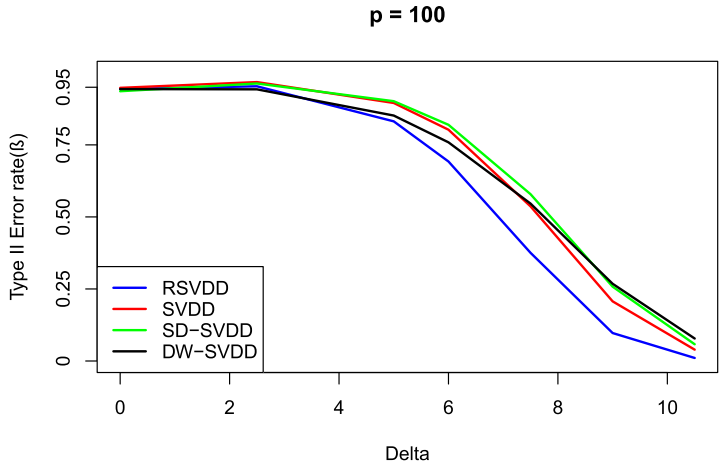

Figure 3: Type II error plot for $p = 100$.

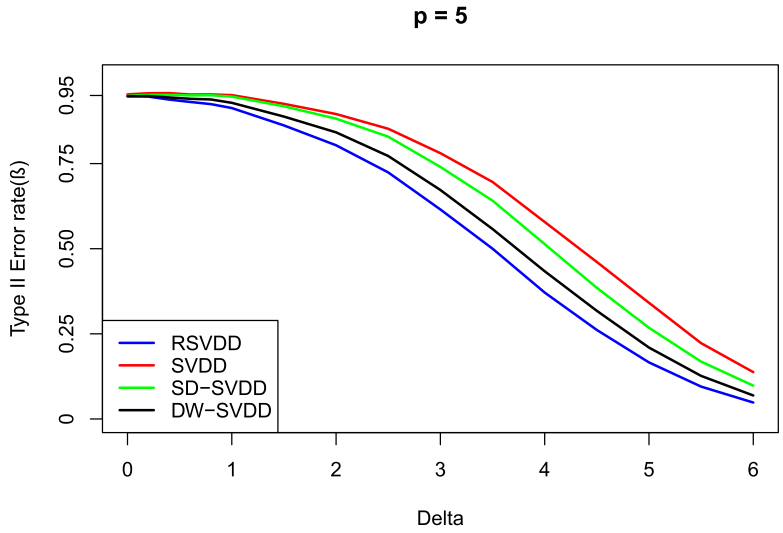

Figure 4: Type II error plot for $p = 5$.

4.4 Correlated Simulation Results and Discussion

The results from the correlated multivariate normal settings for both $p = 5$ and $p = 10$ shows that all models achieve lower Type II error rates (β) as the mean shift parameter δ increases, which aligns with expectations—larger shifts from the target distribution facilitate more effective anomaly detection. Across both settings, RSVDD demonstrates the most favorable performance, maintaining the lowest Type II error rates at nearly all δ levels. This underscores its robustness to feature correlation and higher dimensionality. In contrast, SVDD yields the highest Type II errors, indicating that it is less effective in correlated data environments. Stahel–Donoho SVDD (SD-SVDD) and Density-Weighted SVDD (DW-SVDD) show moderate performance. While both outperform SVDD, SD-SVDD tends to perform slightly better than DW-SVDD, particularly at moderate shift levels. However, their relative performance narrows as δ increases. A noticeable trend is that at small δ values, where the detection task is most difficult, the performance gap between RSVDD and the other methods is most pronounced. As δ grows, all models approach lower error rates, though RSVDD maintains a consistent advantage. When moving from $p = 5$ to $p = 10$, there is a slight increase in Type II error at lower δ values across all models, suggesting that higher dimensionality introduces additional challenges for anomaly detection. Nonetheless, RSVDD continues to perform reliably and better than the other approaches, confirming its scalability and effectiveness in more complex feature spaces. Overall, these findings support the conclusion that RSVDD is the most robust and accurate method for detecting anomalies in correlated, high-dimensional settings, while SVDD remains the most sensitive to correlation and dimensionality.

4.4.1 Model Runtimes (Seconds) for Uncorrelated and Correlated Settings

All experiments were implemented in R version 4.3.2 and executed on a Windows 11 Home 64-bit system equipped with an Intel Core i7-1065G7 CPU (8 cores, 1.5 GHz) and 12 GB of RAM. As shown in Tables 1 and 2, RSVDD exhibits longer training times compared to SVDD, DS-SVDD, and DW-SVDD. This is expected since RSVDD relies on an iterative optimization scheme, while

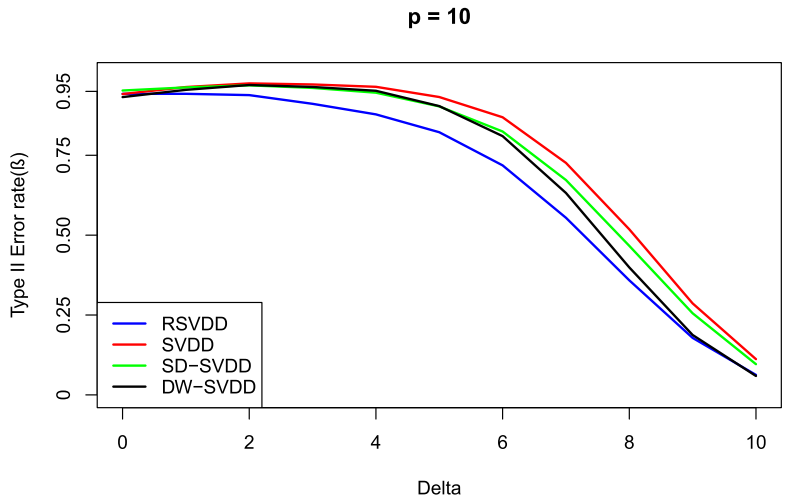

Figure 5: Type II error plot for $p = 10$.

Table 1: Computational time (uncorrelated setting).

Method	$p = 5$	$p = 10$	$p = 100$
RSVDD	11.90	6.99	2.02
SVDD	0.06	0.07	0.05
DS-SVDD	0.57	0.46	0.07
DW-SVDD	0.08	0.06	0.02

Table 2: Computational time (correlated setting).

Method	$p = 5$	$p = 10$
RSVDD	8.10	7.65
SVDD	0.09	0.07
DS-SVDD	0.45	0.41
DW-SVDD	0.08	0.08

SVDD and its variants solve the optimization problem via standard quadratic programming, which is generally more computationally efficient, particularly in lower dimensions. Among the kernel-based methods, DW-SVDD consistently records the shortest training time. This can be attributed to its density-weighted formulation, which simplifies the optimization landscape and reduces computational burden. Despite being computationally more intensive, RSVDD scales more efficiently with increased dimensionality and demonstrates robust performance, making it a competitive choice for anomaly detection in higher-dimensional settings. These findings highlight a trade-off between robustness and computational efficiency: while RSVDD incurs higher training costs, it offers improved robustness properties, especially in contaminated settings.

5 Illustrative Example

The application of RSVDD is demonstrated through its ability to accurately distinguish between data points representing the target class in the Breast Cancer Dataset, which was developed by Wolberg and Mangasarian (1990). This dataset includes two distinct target classes, identified as 2 or 4. It comprises nine features collected from 683 patients: 444 of these patients were diagnosed with benign tumors (target class), and 239 were diagnosed with malignant tumors (non-target class). For training purposes, we selected the first 75 samples from the benign tumor group as primary points and 5 samples from the malignant group as outlying data points, creating a training dataset with $p = 9$ variables and $N = 80$ observations.

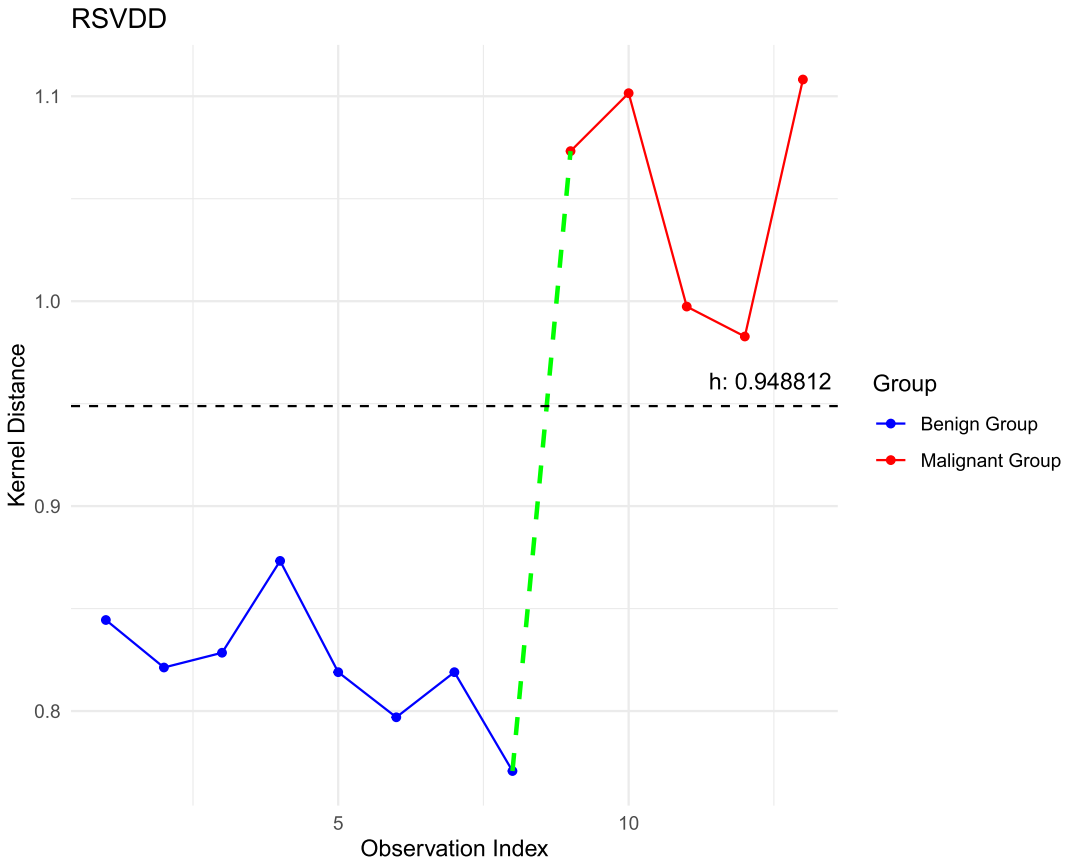

Figure 6: OCC using RSVDD for the breast cancer dataset example.

We have a training sample of 9 component vectors x_j, $j = 1, 2, \ldots, N$ assumed to be in the primary class. The RSVDD problem is to solve the problem discussed in equation (22) above.

By using Algorithms 1 and 2, we have the following results:

The Breast Cancer example in Figure 6–9 depicts plots representing kernel distance measurements for benign and malignant groups, utilizing RSVDD, SVDD, SD-SVDD, and DW-SVDD methods. The RSVDD plot shows a clear demarcation between benign (blue) and malignant (red) observations, with all data points appropriately classified below or above the threshold. Conversely, the SVDD plot reveals a notable misclassification where one benign data point (blue) surpasses the threshold significantly, falsely indicating a malignant characteristic around the 4th observation index. Similarly, in SD-SVDD and DW-SVDD, the 9th point is also misclassified. These misclassifications illustrate potential limitations or sensitivities in the SVDD, SD-SVDD, and DW-SVDD approach compared to RSVDD, which appears more robust and reliable in this scenario, maintaining strict adherence to the established threshold for each group.

6 Conclusion

In this study, we have introduced the Rescale Hinge Loss Support Vector Data Description (RSVDD), an innovative extension of the SVDD model designed to enhance its robustness against anomalies and outliers. By incorporating a rescaled hinge loss function and leveraging

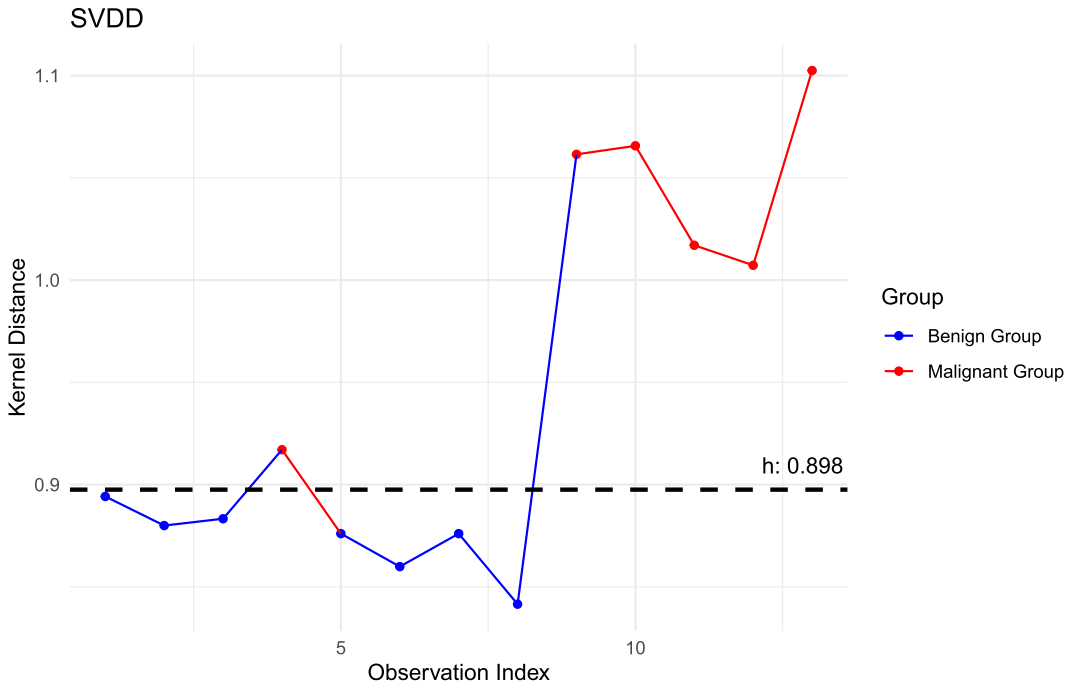

Figure 7: OCC using SVDD for the breast cancer dataset example.

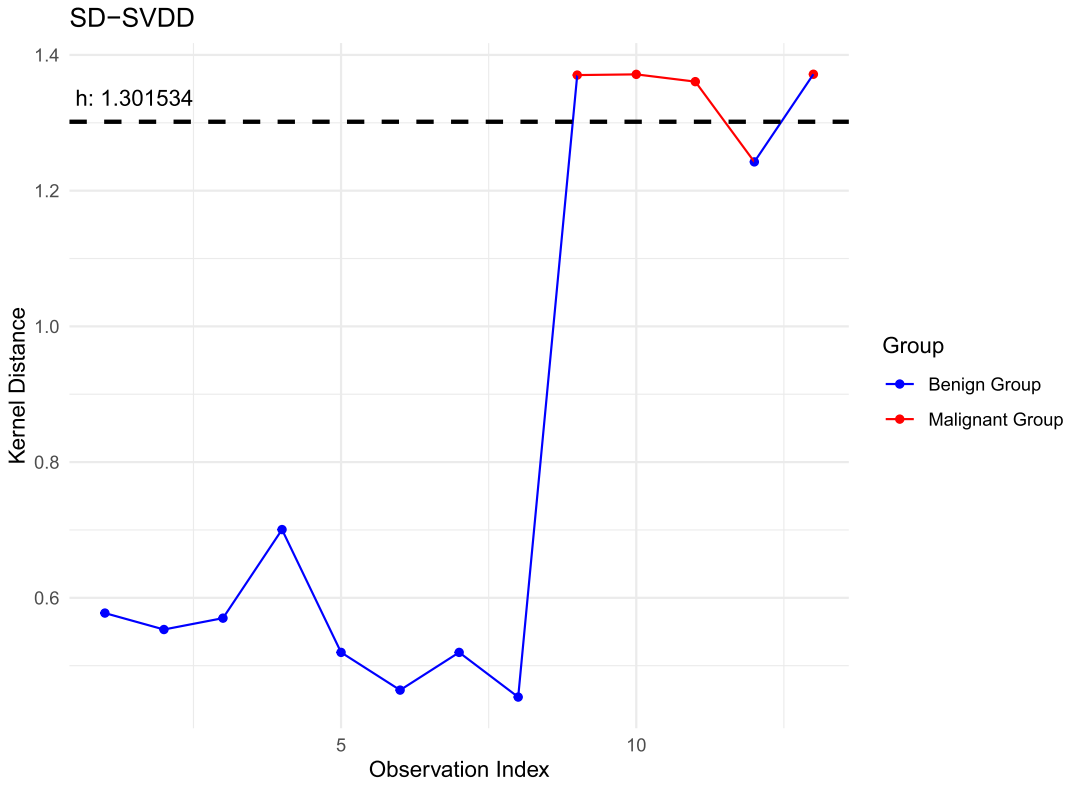

Figure 8: OCC using SD-SVDD for the breast cancer dataset example.

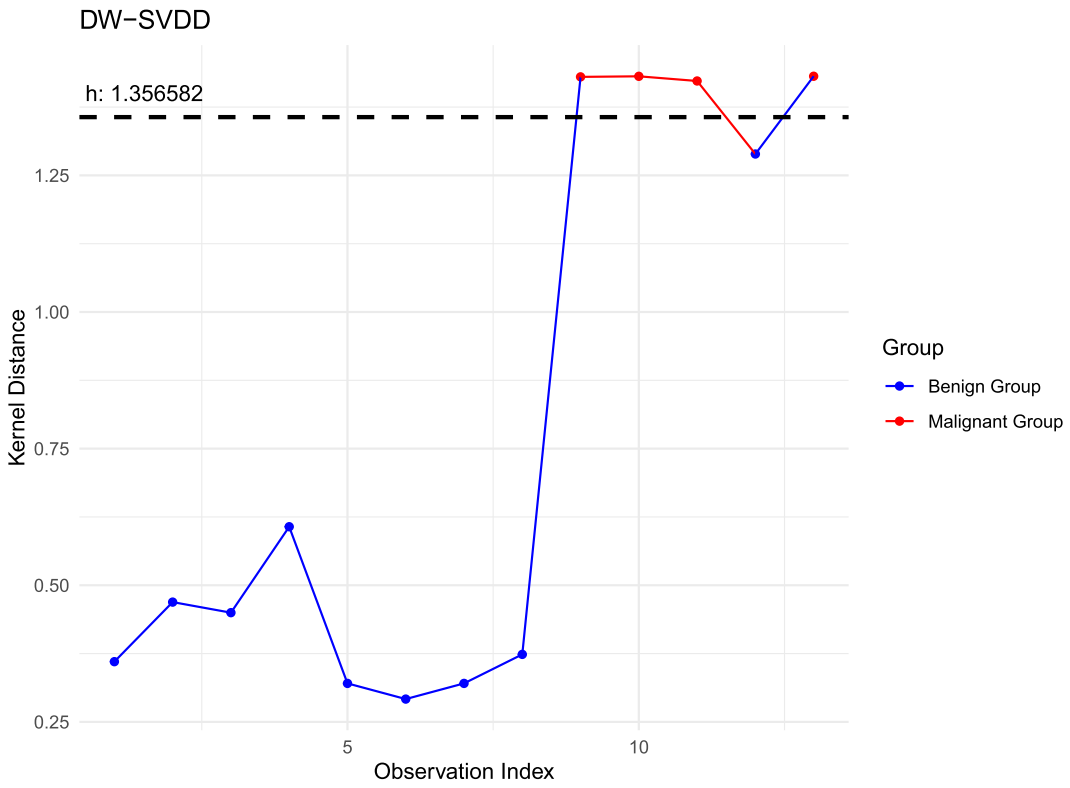

Figure 9: OCC using DW-SVDD for the breast cancer dataset example.

the half-quadratic optimization method, RSVDD demonstrates superior performance in anomaly detection tasks, especially when dealing with contaminated data.

Our experimental results, validated through synthetic and breast cancer datasets, show that RSVDD consistently outperforms standard SVDD, Density-Weighted SVDD (DW-SVDD), and Stahel-Donoho SVDD (SD-SVDD) across various metrics. RSVDD's ability to maintain lower Type II error rates highlights its effectiveness and reliability in practical applications, where the presence of outliers can significantly impact model performance.

The robust nature of RSVDD, coupled with its dynamic optimization algorithm, positions it as a valuable tool in one-class classification systems, offering significant improvements in accuracy and robustness. This advancement highlights the potential of integrating sophisticated loss functions and optimization techniques in developing robust machine learning models.

Future research could explore further enhancements to the RSVDD framework, such as extending the model to handle multi-class classification tasks. Overall, RSVDD sets a new standard in anomaly detection, contributing to the ongoing evolution of robust machine learning methodologies.

Supplementary Material

We have provided all the supplementary materials necessary to successfully reproduce this work, including the simulation data, corresponding code, and illustrative examples.

A Appendix

Let

$$l(u) = vu + u\log(-u) - u. \tag{30}$$

Then, we have

$$g^*(v) = \operatorname*{Sup}_{u<0}\{l(u)\}. \tag{31}$$

We then find the derivative of $l(u)$ concerning u and set it to 0.

$$\begin{aligned}\frac{\partial l}{\partial u} &= v + u\left(-\frac{1}{u}(-1)\right) + \log(-u) - 1\\ &= v + 1 + \log(-u) - 1\\ &= v + \log(-u) = 0\\ &= \log(-u) = -v\\ &= -u = \exp\{-v\}\\ &= u = -\exp\{-v\} < 0.\end{aligned}$$

Now, we let $v = \eta l_{\text{hinge}}(z_j)$ and $u^* = -\exp\{-\eta l_{\text{hinge}}(z_j)\}$.

By substituting u^* and v into (32) we have

$$\begin{aligned}l(u^*) &= -\eta l_{\text{hinge}}(z_j)\exp\{-\eta l_{\text{hinge}}(z_j)\} - \exp\{-\eta l_{\text{hinge}}(z_j)\}\\ &\quad\times\log\big(-(-\exp\{-\eta l_{\text{hinge}}(z_j)\})\big) + \exp\{-\eta l_{\text{hinge}}(z_j)\}\\ &= -\eta l_{\text{hinge}}(z_j)\exp\{-\eta l_{\text{hinge}}(z_j)\} + \eta l_{\text{hinge}}(z_j)\\ &\quad\times\exp\{-\eta l_{\text{hinge}}(z_j)\} + \exp\{-\eta l_{\text{hinge}}(z_j)\}\\ &= \exp\{-\eta l_{\text{hinge}}(z_j)\}.\end{aligned}$$

Hence, we derive that

$$\begin{aligned}g^*(v) &= \operatorname*{Sup}_{\mathbf{u}<0}\{vu + u\log(-u) - u\}\big|_{u=-\exp\{-\eta l_{\text{hinge}}(z_j)\}}\\ &= \exp\{-\eta l_{\text{hinge}}(z_j)\},\end{aligned}$$

where the supremum is achieved at $u = -\exp\{-\eta l_{\text{hinge}}(z_j)\} < 0$.

Therefore, we can set $\operatorname{Sup}_{\mathbf{u}<0}\{\eta l_{\text{hinge}}(z_j)u - g(u)\} = \exp\{-\eta l_{\text{hinge}}(z_j)\}$.

B Appendix

Half-Quadratic (HQ) Optimization for the Solution of RSVDD

Definition. Let $f : \mathbb{R}^p \to \mathbb{R}$. The function $f^* : \mathbb{R}^p \to \mathbb{R}$, defined as

$$f^*(\mathbf{y}) = \operatorname*{Sup}_{\mathbf{x}}(\mathbf{y}^T\mathbf{x} - f(\mathbf{x})),$$

is called the Fenchel conjugate (or conjugate) of the function f. The advantage of using the Fenchel conjugate is that f^* is bounded above and also a convex function whether the original f is convex or not convex, since f^* is the pointwise supremum of a family of convex (indeed,

affine) functions of $\mathbf{y}$. The fact that f^* is bounded above allows us to alternately optimize our objective function.

To effectively use HQ optimization, we define the following convex function:

$$g(u) = -u\log(-u) + u, \quad u < 0.$$

By conjugate function theory (Boyd and Vandenberghe, 2004), we can express the Fenchel conjugate function $g^*(v)$ of $g(u)$ as

$$g^*(v) = \operatorname*{Sup}_{u<0}\{vu + u\log(-u) - u\}, \tag{32}$$

where the supremum is achieved at $u = -\exp\{-\eta l_{\text{hinge}}(z_j)\} < 0$.

Consequently, we derived that

$$g^*(v) = \operatorname*{Sup}_{u<0}\{\eta l_{\text{hinge}}(z_j)u - g(u)\}\big|_{u=-\exp\{-\eta l_{\text{hinge}}(z_j)\}} \tag{33}$$

$$= \exp\{-\eta l_{\text{hinge}}(z_j)\}. \tag{34}$$

The derivation of the supremum value of u and (34) can be found in Appendix A. Now, from (34), we can rewrite $\mathcal{L}_{l2}(R, \mathbf{a})$ in (26) in the following way

$$\begin{aligned}\mathcal{L}_{l2}(R, \mu) &= -R^2 + C\beta\sum_{j=1}^{N}\sup_{u_j<0}\{-\eta l_{\text{hinge}}(z_j)u_j - g(u_j)\}\\ &= -R^2 + C\beta\sup_{\mathbf{u}<0}\left\{\sum_{j=1}^{N} -\eta l_{\text{hinge}}(z_j)u_j - g(u_j)\right\}\\ &= \sup_{\mathbf{u}<0}\left\{-R^2 + C\beta\sum_{j=1}^{N}\{\eta l_{\text{hinge}}(z_j)u_j - g(u_j)\}\right\},\end{aligned} \tag{35}$$

where $\mathbf{u} \in \mathbb{R}^N$ and $u_j < 0$.

C Appendix

Proof of Proposition 2

Proof. Comparing (35) and (23), we know that $\mathcal{L}_{l3}(R, \mu, \mathbf{u}) \leqslant \mathcal{L}_{l2}(R, \mu) \leqslant C\beta$. That is to say $\mathcal{L}_{l3}(R, \mu, \mathbf{u})$ is upper bounded. Then we can deduce from (30) and (32) that $\mathcal{L}_{l3}(R^\tau, \mu^\tau, \mathbf{u}^\tau) \leqslant \mathcal{L}_{l3}(R^{\tau+1}, \mu^{\tau+1}, \mathbf{u}^\tau) \leqslant \mathcal{L}_{l3}(R^{\tau+1}, \mu^{\tau+1}, \mathbf{u}^{\tau+1})$. Therefore the sequence $\{\mathcal{L}_{l3}(R^\tau, \mu^\tau, \mathbf{u}^\tau), \tau = 1, 2, \dots\}$ is non-decreasing. Hence, we verify that $\{\mathcal{L}_{l3}(R^\tau, \mu^\tau, \mathbf{u}^\tau), \tau = 1, 2, \dots\}$ of Algorithm 1 converges. □

Funding

This work is partially supported by Microsoft.

References

Boyd S, Vandenberghe L (2004). *Convex Optimization*. Cambridge University Press.

Cha M, Kim JS, Baek JG (2014). Density weighted support vector data description. *Expert Systems with Applications*, 41(7): 3343–3350. https://doi.org/10.1016/j.eswa.2013.11.025

Donoho DL (1982). Breakdown properties of multivariate location estimators. *Technical report*, Harvard University, Boston. http://www-stat.stanford.edu/~

Erfani SM, Rajasegarar S, Karunasekera S, Leckie C (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58: 121–134. https://doi.org/10.1016/j.patcog.2016.03.028

Ghasemi E, Shahbahrami A, Hashemi M (2021). Robust support vector data description based on information entropy for one-class classification. *Expert Systems with Applications*, 170: 114403.

Hu W, Hu T, Wei Y, Lou J, Wang S (2021). Global plus local jointly regularized support vector data description for novelty detection. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9): 6602–6614. https://doi.org/10.1109/TNNLS.2021.3129321

Kennedy K, Mac Namee B, Delany SJ (2009). Learning without default: A study of one-class classification and the low-default portfolio problem. In: Bridge D, Doyle D, Hayes P (Eds.), *Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science*, 174–187. Springer.

Khan NM, Ksantini R, Ahmad IS, Guan L (2014). Covariance-guided one-class support vector machine. *Pattern Recognition*, 47(6): 2165–2177. https://doi.org/10.1016/j.patcog.2014.01.004

Khan SS, Madden MG (2014). One-class classification: Taxonomy of study and review of techniques. *Knowledge Engineering Review*, 29(3): 345–374. https://doi.org/10.1017/S026988891300043X

Kivinen J, Smola AJ, Williamson RC (2004). Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8): 2165–2176. https://doi.org/10.1109/TSP.2004.830991

Liu W, Pokharel PP, Principe JC (2007). Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Transactions on Signal Processing*, 55(11): 5286–5298. https://doi.org/10.1109/TSP.2007.896065

Maboudou-Tchao EM (2018). Kernel methods for changes detection in covariance matrices. *Communications in Statistics. Simulation and Computation*, 47(6): 1704–1721. https://doi.org/10.1080/03610918.2017.1322701

Maboudou-Tchao EM (2020). Change detection using least squares one-class classification control chart. *Quality Technology & Quantitative Management*, 17(5): 609–626. https://doi.org/10.1080/16843703.2019.1711302

Maboudou-Tchao EM (2021a). High-dimensional data monitoring using support machines. *Communications in Statistics. Simulation and Computation*, 50(7): 1927–1942. https://doi.org/10.1080/03610918.2019.1588312

Maboudou-Tchao EM (2021b). Monitoring the mean with least-squares support vector data description. *Gestão & Produção*, 28(3): e019. https://doi.org/10.1590/1806-9649-2021v28e019

Maboudou-Tchao EM (2021c). Support tensor data description. *Journal of Quality Technology*, 53(2): 109–134. https://doi.org/10.1080/00224065.2019.1642815

Maboudou-Tchao EM (2023). Least-squares support tensor data description. *Communications in Statistics. Simulation and Computation*, 52(7): 3026–3042. https://doi.org/10.1080/03610918.2021.1926500

Maboudou-Tchao EM, Hampton HD (2025). Deep least squares one-class classification. *Journal of Quality Technology*, 57(1): 68–92. https://doi.org/10.1080/00224065.2024.2421164

Maboudou-Tchao EM, Harrison CW (2021). A comparative study of L_1 and L_2 norms in support vector data descriptions. In: Chatterjee S, Sarker R, Herrmann JW (Eds.), *Control Charts and Machine Learning for Anomaly Detection in Manufacturing*, 217–241. Springer.

Maboudou-Tchao EM, Silva IR, Diawara N (2018). Monitoring the mean vector with Mahalanobis kernels. *Quality Technology & Quantitative Management*, 15(4): 459–474. https://doi.org/10.1080/16843703.2016.1226707

Nikolova M, Ng MK (2005). Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific Computing*, 27(3): 937–966. https://doi.org/10.1137/030600862

Principe JC (2010). *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Springer Science & Business Media.

Ruff L, Vandermeulen R, Goernitz N, Deecke L, Siddiqui SA, Binder A, et al. (2018). Deep one-class classification. In: Dy J, Krause A (Eds.), *Proceedings of the 35th International Conference on Machine Learning*, 4393–4402. PMLR.

Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7): 1443–1471. https://doi.org/10.1162/089976601750264965

Schölkopf B, Williamson RC, Smola A, Shawe-Taylor J, Platt J (1999). Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, 12.

Seliya N, Abdollah Zadeh A, Khoshgoftaar TM (2021). A literature review on one-class classification and its potential applications in big data. *Journal of Big Data*, 8: 1–31. https://doi.org/10.1186/s40537-020-00387-6

Singh A, Pokharel R, Principe J (2014). The c-loss function for pattern classification. *Pattern Recognition*, 47(1): 441–453. https://doi.org/10.1016/j.patcog.2013.07.017

Stahel WA (1981). Robust estimation: Infinitesimal optimality and covariance matrix estimators. *Unpublished doctoral dissertation*, ETH, Zurich, Switzerland.

Sun R, Tsung F (2003). A kernel-distance-based multivariate control chart using support vector methods. *International Journal of Production Research*, 41(13): 2975–2989. https://doi.org/10.1080/1352816031000075224

Tax DM, Duin RP (1999). Data domain description using support vectors. In: *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 251–256.

Tax DM, Duin RP (2004). Support vector data description. *Machine Learning*, 54: 45–66. https://doi.org/10.1023/B:MACH.0000008084.60811.49

Wang CD, Lai J (2013). Position regularized support vector domain description. *Pattern Recognition*, 46(3): 875–884. https://doi.org/10.1016/j.patcog.2012.09.018

Wang K, Lan H (2020). Robust support vector data description for novelty detection with contaminated data. *Engineering Applications of Artificial Intelligence*, 91: 103554. https://doi.org/10.1016/j.engappai.2020.103554

Wolberg WH, Mangasarian OL (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87(23): 9193–9196. https://doi.org/10.1073/pnas.87.23.9193

Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y (2008). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2): 210–227. https://doi.org/10.1109/TPAMI.2008.79

Xu G, Cao Z, Hu BG, Principe JC (2017). Robust support vector machines based on the rescaled hinge loss function. *Pattern Recognition*, 63: 139–148. https://doi.org/10.1016/j.patcog.2016.09.045

Yang Y, Cheng X, Gao Y (2017). Enhancing robustness in SVDD for imbalanced and noisy data classification. *Applied Intelligence*, 47(4): 1066–1078.