Pattern Recognition 110 (2021) 107648

Contents lists available at ScienceDirect

Pattern Recognition

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

Multimodal subspace support vector data description

Fahad Sohrab[a,*], Jenni Raitoharju[a,b], Alexandros Iosifidis[c], Moncef Gabbouj[a]

[a] Faculty of Information Technology and Communication Sciences, Tampere University, FI-33720 Tampere, Finland
[b] Programme for Environmental Information, Finnish Environment Institute, FI-40500 Jyväskylä, Finland
[c] Department of Engineering, Electrical and Computer Engineering, Aarhus University, DK-8200 Aarhus, Denmark

ARTICLE INFO

Article history:
Received 21 August 2019
Revised 13 July 2020
Accepted 6 September 2020
Available online 10 September 2020

Keywords:
Feature transformation
Multimodal data
One-class classification
Support vector data description
Subspace learning

ABSTRACT

In this paper, we propose a novel method for projecting data from multiple modalities to a new subspace optimized for one-class classification. The proposed method iteratively transforms the data from the original feature space of each modality to a new common feature space along with finding a joint compact description of data coming from all the modalities. For data in each modality, we define a separate transformation to map the data from the corresponding feature space to the new optimized subspace by exploiting the available information from the class of interest only. We also propose different regularization strategies for the proposed method and provide both linear and non-linear formulations. The proposed Multimodal Subspace Support Vector Data Description outperforms all the competing methods using data from a single modality or fusing data from all modalities in four out of five datasets.

1. Introduction

In our surroundings, on a daily basis, we are exposed to information from many different sources. Different sensors are used to gather information about similar objects. Our brains usually perform well in combining the information from different sources to make a concise analysis of that particular entity. In order to analyze an entity, even a single source of information might be enough, but to make some critical decisions it is important to combine information from different sources in a systematic way. For example, if a person is walking in a crowd, the main information to not hit anything comes from visual cues, but people can warn each other also by voice or even by touch, and this extra information helps in understanding the environment in a better way. The smell could help to avoid unpleasant spots, too. As another example, while watching a movie, only visual information of the scenes may not be enough to understand the whole scenario, but the audio and/or captions combined together with the visuals information will provide the full information.

In machine learning techniques for predictive data modeling, training data are used to form a model that can accurately classify future instances into a predefined number of classes. In many cases, data comes from sensors and can be further processed to extract different features. The term *multimodal* is used to describe the data coming from different sensors (also referred to as mode or modality), however, it is also used as a synonym to *multi-view* when different features are extracted from the same sensor or when there are multiple similar sensors, e.g., cameras. The aim of multimodal machine learning algorithms is to build models that can process and relate information from more than one modality (or view).

The examples of multimodal representations are prevalent in different application areas. In [1], an active multimodal sensor system for target recognition and tracking is studied where information from three different sensors (visual, infrared, and hyperspectral) is used. In [2], a framework for vehicle tracking with multimodal data (velocity and images) is proposed where the outcome of velocity modality estimated by using a Kalman filter on the data obtained from motion sensors is fused with features learned from image modality by the color-faster R-CNN method. In [3], a multimodal data collection framework for mental stress monitoring is studied. In the proposed framework, physiological and motion sensor data of people under stress are collected.

* Corresponding author.
E-mail addresses: fahad.sohrab@tuni.fi (F. Sohrab), jenni.raitoharju@tuni.fi (J. Raitoharju), alexandros.iosifidis@eng.au.dk (A. Iosifidis), moncef.gabbouj@tuni.fi (M. Gabbouj).

https://doi.org/10.1016/j.patcog.2020.107648

The data in multimodal applications come from different modalities, where each modality has its own statistical properties and contains specific information. The different modalities usually share high-level concepts and semantic information, and all together contain more information than any single-modal data [4]. If we build a model separately for each modality, the relationship between the modalities cannot be exploited efficiently. In multimodal subspace learning, the goal is to infer a shared latent representation, that can accurately model data from each original modality and exploit the relationship between the modalities.

In traditional multiclass machine learning, an adequate amount of data are available for all the categories during training and, hence, the algorithm takes advantage of all available training data from all classes to train a model [5]. However, it is possible that during the training, data are highly imbalanced, or the only data available is from a single class. In such cases, one-class classification techniques are used. It is useful in many different cases, such as outlier detection, predicting specific events, or, in general, predicting a specific target class. While much effort has been put on solving one-class classification tasks for data of a single modality [6], much less effort has been put on solving one-class multimodal challenges in general, and we are not aware of any prior work in the field of multimodal learning for one-class classification. In one-class multimodal tasks, it is assumed that the only data available is from a single class in many different modalities.

In this paper, we propose a novel method for solving multimodal one-class classification tasks. The proposed method, Multimodal Subspace Support Vector Data Description (MS-SVDD), finds a transformation for each modality along with defining a common model for all modalities in a lower-dimensional subspace optimized for one-class classification. The rest of the paper is organized as follows. In Section 2, an overview of related work is presented. In Section 3, the newly proposed MS-SVDD is derived and discussed. In Section 4, we present the experimental setup and results, and finally, in Section 5, conclusions are drawn.

2. Background and related work

In this section, we briefly discuss the principles of multimodal learning, along with subspace learning. We also provide an overview of traditional methods used for multiclass multimodal data description and one-class unimodal data description.

2.1. Multimodal learning

The availability of many different modalities can be bliss if it increases the performance of the machine learning model. However, if the data description algorithm fails to make a strong connection between the different available modalities, the performance can be degraded. To ensure better performance of the model by combining data from different modalities, mainly two principles should be ensured, i.e., consensus and complementary principles [7]:

- **Consensus principle** aims at minimizing the disagreement between data available from different modes. Maximizing the agreement will reduce the error rate, and better modeling of data is achieved while combining data from different modalities.
- **Complementary principle** in the context of multimodal learning means that data from each modality may contain some knowledge not contained by the other ones. So it is necessary to exploit information from all the available modes to make an accurate description of data.

The multimodal machine learning techniques can be described by three main properties: two-view vs. multi-view, linear vs. non-linear, and unsupervised vs. supervised [8]. As the name indicates, in two-view learning, the number of views is limited to two. In multi-view learning, the number of views is not limited. The difference between supervised and unsupervised learning is that, in supervised learning, the information on output labels of the training data is taken into account when training the model, while in unsupervised methods, the labels are not used to model the underlying structure or distribution of the data [9]. Linear techniques for multimodal subspace learning may be too simple to provide a representative model. Hence, kernel methods are proposed to capture non-linear patterns in data.

The multimodal learning techniques have been mainly applied on four applications domains [10]: i.e., audio-visual speech recognition [11], multimedia content indexing and retrieval [12], understanding human multimodal behaviors [13], and language and vision media description [14]. Recently, there has been a rising trend in applying multimodal machine learning algorithms also to other applications. For example, in [15], a multimodal data fusion technique is used for the prediction of soybean yield from an unmanned aerial vehicle.

In multimodal learning, the main goal is to develop a process of fusing information from various modalities. In [16], the fusion strategies are divided into two different categories as model-agnostic and model-based approaches. In model-agnostic approaches, the fusion is either late, early, or hybrid. In early fusion, the data or extracted features are fused together at the very initial phase of modeling. A new feature vector is usually formed by concatenating all the available data from different modes, and the model is trained with the new feature vector. In late fusion, multiple models are trained, and the fusion is done for scores generated by each model for the corresponding modality. The score generated by each model can be a threshold or some probability used in decision making. Hybrid fusion exploits the advantage of both early fusion and late fusion. Model-based approaches for fusion explicitly fuses data during their construction, such as kernel-based approaches, graphical models, and neural networks. In this work, we present a model-based approach for data fusion.

2.2. Subspace learning

In the current era of data science, where high-dimensional multimodal big data are generated every minute in different industries, there is a need to get the essential insights and mine knowledge in this high-dimensional data. Subspace learning aims at representing data in a lower-dimensional space by keeping intact all the information available in the original higher-dimensional space.

Algorithms developed for linear subspace learning find a projection matrix for labeled training data (represented by vectors) satisfying some optimality criteria. Principal Component Analysis (PCA) is one of the first subspace learning methods mentioned in literature. In PCA, a subspace is learned by orthogonally projecting data to a subspace so that the variance of data is maximized. PCA works only with a single mode of data, i.e., all data should be in the same dimension. Another traditional subspace learning method is Linear Discriminant Analysis (LDA), which finds a linear transformation by exploiting the class information.

Analogous to PCA, but used for two-view learning, is canonical-correlation analysis (CCA) [17]. CCA is a classic and conventional method for subspace learning, which aims at relating two sets of data by finding out the pairs of directions which provide a maximum correlation between the two sets. It has recently become one of the popular methods for unsupervised subspace learning because of its generalization capability and has been used extensively for multimodal data fusion and cross-media retrieval [18]. In subspace learning, state-of-the-art results are achieved by methods which have embraced some stimulus from conventional subspace learning methods [19].

As an extension of methods for linear transformation, kernel methods are introduced to describe nonlinear function or decision boundaries. In kernel methods, the data are mapped to a typically higher-dimensional kernel-space using a kernel function where it exhibits linear patterns [20,21]. For example, in [22], kernel-PCA performing a nonlinear form of PCA is proposed.

2.3. One-class classification

In one-class classification, the parameters of the model are estimated using data from the positive class only because data from the other classes are either not available at all or it is too diverse in nature to be modeled statistically [23]. The positive class is also called the target class, and the data from the other classes, which are not available during training, is called negative, or an outlier class. For example, a unimodal biometric system uses a single biometric trait for verification or identification [24].

Support Vector Data Description (SVDD) [25] is among the most widely used one-class classification methods used for anomaly detection and other related applications. SVDD obtains a spherical boundary around target data which can be made flexible by using the kernel trick. The obtained boundary is used to detect outliers during the test, i.e., anything inside the closed boundary is classified as a target class and otherwise as an outlier. The Lagrangian of SVDD is given as follows

$$L = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i^T \mathbf{x}_i - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \mathbf{x}_i^T \mathbf{x}_j \alpha_j, \tag{1}$$

where $\mathbf{x}_i$ is the input target training instance and maximizing (1) gives a set of α_i corresponding to each instance. The instances with $\alpha_i \geq 0$ define the data description. Other common one-class classification method is One-Class Support Vector Machine (OC-SVM) [26].

Techniques for enhancing the performance of one-class classification methods, mainly extensions of SVDD, can be categorized into four main categories: methods based on data structure, kernel issue, boundary shape, and non-stationary data [27]. As the name indicates, in the data structure category, the main focus is on the structure of data. For example, in [28], a confidence coefficient is associated with each training sample to deal with the uncertainty of data. In kernel issue extensions, the main focus is on reducing the complexity or proposing new kernels for one-class classification. For example, in [29], a new kernel is proposed to improve the accuracy of SVDD for time series classification. Proposing changes in the boundary for enclosing the target data comes under the third category for improving one-class classification accuracy. For example, in [30], the ellipse shape is used for encapsulating target data instead of the traditional sphere used in SVDD. In [31], it is shown that both SVDD and OC-SVM lead to the same solution when exploiting the elliptical shape of the class. The last category of algorithms for improving one-class classifier performance attempts to handle non-stationary data. For example in [32], Incremental-SVDD (I-SVDD) is proposed to handle non-stationary or increasing data. Recently, in [33], an algorithm developed for reducing the effect of uncertain data around the hypersphere of SVDD achieved the state of the art result on many UCI [34] datasets. In this paper, we consider baseline SVDD combined with multimodal subspace learning. However, in the future, the method can be further extended using similar ideas.

In the area of multimodal one-class classification, researchers have mainly focused on fusing the output labels of multiple models trained for each type of feature independently, i.e., without taking into account information from other feature types for one model [35].

3. Multimodal subspace support vector data description

MS-SVDD maps data from high-dimensional feature spaces to a low-dimensional feature space optimized for one-class classification. The optimized subspace is shared by data coming from all modalities. MS-SVDD is an extension of Subspace Support Vector Data Description (S-SVDD), which was proposed for unimodal data in [36]. The main novelty of MS-SVDD is using the multimodal approach for one-class classification. Here, we first derive the linear MS-SVDD. Then we derive two non-linear versions using the kernel trick [20] and the Nonlinear Projection Trick (NPT) [37], respectively.

3.1. Linear MS-SVDD

Let us assume that the items to be modelled are represented by M different modalities. The instances in each modality m, $m = 1, \ldots, M$, are represented by $\mathbf{X}_m = [\mathbf{x}_{m,1}, \mathbf{x}_{m,2}, \ldots \mathbf{x}_{m,N}]$, $\mathbf{x}_{m,i} \in \mathbb{R}^{D_m}$, where N is the total number of instances and D_m is the dimensionality of the feature space in modality m. MS-SVDD tries to find a projection matrix $\mathbf{Q}_m \in \mathbb{R}^{d \times D_m}$ for each modality, which will project the corresponding instances to a lower (d)-dimensional optimized subspace shared by all modalities. Thus, a feature vector $\mathbf{x}_{m,i}$ is projected to a d-dimensional vector $\mathbf{y}_{m,i}$ as

$$\mathbf{y}_{m,i} = \mathbf{Q}_m \mathbf{x}_{m,i}, \forall m \in \{1, \ldots, M\}, \ \ \forall i \in \{1, \ldots, N\}. \tag{2}$$

To obtain a common description of all the data transformed from their corresponding modalities to the new common subspace, we exploit Support Vector Data Description (SVDD) [25] to form a closed boundary around the target class data in the new subspace. The center and radius of the hypersphere are denoted by $\mathbf{a} \in \mathbb{R}^d$ and R, respectively. Fig. 1 depicts the basic idea of the proposed method.

In order to find a compact hypersphere which encloses all the target data from all the modalities in the new subspace, we minimize

$$F(R, \mathbf{a}) = R^2$$

s.t.

$$\|\mathbf{Q}_m \mathbf{x}_{m,i} - \mathbf{a}\|_2^2 \leq R^2, \forall m \in \{1, \ldots, M\}, \ \ \forall i \in \{1, \ldots, N\}. \tag{3}$$

By introducing slack variables $\xi_{m,i}$, such that most of the training data from all the modalities in the new common space should lie inside the hypersphere, the above criterion becomes

$$F(R, \mathbf{a}) = R^2 + C \sum_{m=1}^{M}\sum_{i=1}^{N} \xi_{m,i}$$

s.t.

$$\|\mathbf{Q}_m \mathbf{x}_{m,i} - \mathbf{a}\|_2^2 \leq R^2 + \xi_{m,i}, \xi_{m,i} \geq 0, \\ \forall m \in \{1, \ldots, M\}, \forall i \in \{1, \ldots, N\}. \tag{4}$$

The Lagrange function corresponding to (4) can be given as

$$L = R^2 + C \sum_{m=1}^{M}\sum_{i=1}^{N} \xi_{m,i} - \sum_{m=1}^{M}\sum_{i=1}^{N} \gamma_{m,i}\xi_{m,i} - \sum_{m=1}^{M}\sum_{i=1}^{N} \alpha_{m,i}\Big(R^2 + \xi_{m,i} \\ - \mathbf{x}_{m,i}^T \mathbf{Q}_m^T \mathbf{Q}_m \mathbf{x}_{m,i} + 2\mathbf{a}^T \mathbf{Q}_m \mathbf{x}_{m,i} - \mathbf{a}^T \mathbf{a}\Big) \tag{5}$$

The Lagrangian function should be maximized with respect to $\alpha_{m,i} \geq 0$, and $\gamma_{m,i} \geq 0$ and minimized with respect to R, $\mathbf{a}$, $\xi_{m,i}$, and $\mathbf{Q}_m$. By setting the partial derivative to zero, we get

$$\frac{\partial L}{\partial R} = 0 \ \Rightarrow \ \sum_{m=1}^{M}\sum_{i=1}^{N} \alpha_{m,i} = 1 \tag{6}$$

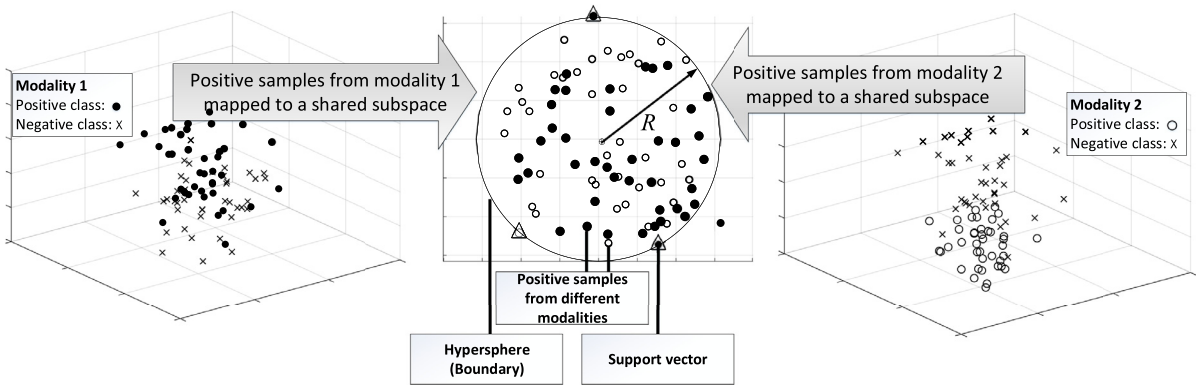

Fig. 1. Depiction of proposed MS-SVDD: Data from two modalities in their corresponding feature space are mapped to a common subspace, where positive class instances are enclosed inside a (hyper)sphere.

$$\frac{\partial L}{\partial \mathbf{a}} = 0 \;\Rightarrow\; \mathbf{a} = \sum_{m=1}^{M}\sum_{i=1}^{N} \alpha_{m,i}\mathbf{Q}_m\mathbf{x}_{m,i} \tag{7}$$

$$\frac{\partial L}{\partial \xi_{m,i}} = 0 \;\Rightarrow\; C - \alpha_{m,i} - \gamma_{m,i} = 0 \tag{8}$$

$$\frac{\partial L}{\partial \mathbf{Q}_m} = 0 \;\Rightarrow\; \mathbf{Q}_m = \Big(\mathbf{a}\sum_{i=1}^{N}\alpha_{m,i}\mathbf{x}_{m,i}^T\Big)\Big(\sum_{i=1}^{N}\alpha_{m,i}\mathbf{x}_{m,i}\mathbf{x}_{m,i}^T\Big)^{-1} \tag{9}$$

It is clear from (6)–(9) that parameters $\boldsymbol{\alpha}$ and $\mathbf{Q}$ are interrelated and cannot be jointly optimized. Hence we apply a two step iterative optimization process where, in each step, we fix one parameter and optimize the other. Substituting (2), (6), (7) and (8) in the Lagrangian function (5), we get

$$L = \sum_{m=1}^{M}\sum_{i=1}^{N}\alpha_{m,i}\mathbf{y}_{m,i}^T\mathbf{y}_{m,i} - \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{n=1}^{M}\sum_{j=1}^{N}\alpha_{m,i}\mathbf{y}_{m,i}^T\mathbf{y}_{n,j}\alpha_{n,j}. \tag{10}$$

We see that optimizing (10) for α corresponds to the traditional SVDD applied in the subspace. Maximizing (10) for a particular set of data will give us $\alpha_{m,i}$ corresponding each sample. The value of $\alpha_{m,i}$ for corresponding sample defines its position with respect to the hypersphere:

- Samples with $0 < \alpha_{m,i} < C$ define the data description and lie on the boundary of hypersphere, they are refered to as support vectors.
- Samples with $\alpha_{m,i} = C$ are outside the boundary.
- Samples with $\alpha_{m,i} = 0$ lie inside the boundary.

In the second step, we fix α and update $\mathbf{Q}_m$ for each modality. For this step, we add a regularization term ω:

$$L = \sum_{m=1}^{M}\sum_{i=1}^{N}\alpha_{m,i}\mathbf{x}_{m,i}^T\mathbf{Q}_m^T\mathbf{Q}_m\mathbf{x}_{m,i} - \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{n=1}^{M}\sum_{j=1}^{N}\alpha_{m,i}\mathbf{x}_{m,i}^T\mathbf{Q}_m^T\mathbf{Q}_n\mathbf{x}_{n,j}\alpha_{n,j} + \beta\omega. \tag{11}$$

The regularization term ω expresses the covariance of data from different modalities in the new low-dimensional space, and β is a regularization parameter for controlling the significance of ω. We propose different settings for ω as

$$\omega_0 = 0, \tag{12}$$

$$\omega_1 = \sum_{m=1}^{M}\mathrm{tr}\big(\mathbf{Q}_m\mathbf{X}_m\mathbf{X}_m^T\mathbf{Q}_m^T\big), \tag{13}$$

$$\omega_2 = \sum_{m=1}^{M}\mathrm{tr}\big(\mathbf{Q}_m\mathbf{X}_m\boldsymbol{\alpha}_m\boldsymbol{\alpha}_m^T\mathbf{X}_m^T\mathbf{Q}_m^T\big), \tag{14}$$

$$\omega_3 = \sum_{m=1}^{M}\mathrm{tr}(\mathbf{Q}_m\mathbf{X}_m\boldsymbol{\lambda}_m\boldsymbol{\lambda}_m^T\mathbf{X}_m^T\mathbf{Q}_m^T), \tag{15}$$

$$\omega_4 = \sum_{m=1}^{M}\sum_{n=1}^{M}\mathrm{tr}(\mathbf{Q}_m\mathbf{X}_m\mathbf{X}_n^T\mathbf{Q}_n^T), \tag{16}$$

$$\omega_5 = \sum_{m=1}^{M}\sum_{n=1}^{M}\mathrm{tr}(\mathbf{Q}_m\mathbf{X}_m\boldsymbol{\alpha}_m\boldsymbol{\alpha}_n^T\mathbf{X}_n^T\mathbf{Q}_n^T), \tag{17}$$

$$\omega_6 = \sum_{m=1}^{M}\sum_{n=1}^{M}\mathrm{tr}(\mathbf{Q}_m\mathbf{X}_m\boldsymbol{\lambda}_m\boldsymbol{\lambda}_n^T\mathbf{X}_n^T\mathbf{Q}_n^T), \tag{18}$$

where $\boldsymbol{\alpha}_m \in \mathbb{R}^N$ in (14) and (17) is a vector having the elements $\alpha_{m,1}, \ldots, \alpha_{m,N}$. Thus, $\boldsymbol{\alpha}_m$ has non-zero values for support vectors and outliers. $\boldsymbol{\lambda}_m \in \mathbb{R}^N$ in (15) and (18) is a vector having the elements of $\boldsymbol{\alpha}_m$ that are smaller than C. Values of $\boldsymbol{\alpha}_m$ corresponding to the outliers (i.e., $\alpha_{m,i} = C$) are replaced with zeros in $\boldsymbol{\lambda}_m$. Thus, $\boldsymbol{\lambda}_m$ has non-zero values only for the support vectors. For ω_0, the regularization term becomes obsolete and it is not used in the optimization process. In ω_1, the regularization term only uses representations coming from the respective modality and no representations from the other modalities are used to describe the variance of the positive class. In ω_2, all support vectors, i.e., representations at the hypersphere boundary, and outliers are used to describe the class variance for the update of the corresponding $\mathbf{Q}_m$. In ω_3, only support vectors of the respective modality are used to describe the variance of the class to be modelled. In ω_4, data from all the modalities are used to describe the covariance and regularize the update of $\mathbf{Q}_m$. In ω_5, the instances belonging to the hypersphere boundary and outliers from all modalities are used to describe the covariance. In ω_6, only the support vectors belonging to class boundary from all modalities are used to update $\mathbf{Q}_m$ and describe the covariance of the positive class.

Note that the MS-SVDD formulation reduces to S-SVDD [36] if data from only one modality ($M = 1$) are taken into account for data description. In S-SVDD, a single projection matrix $\mathbf{Q}$ is determined for mapping the data $\mathbf{X}$ from higher-dimensional space to a lower-dimensional space. A regularization term ψ, which expresses the class variance in the low-dimensional space, is added to the Lagrangian function of S-SVDD:

$$\psi = \mathrm{tr}(\mathbf{Q}\mathbf{X}\boldsymbol{\lambda}\boldsymbol{\lambda}^T\mathbf{X}^T\mathbf{Q}^T), \tag{19}$$

where $\boldsymbol{\lambda}$ can take different forms as described in [36]. The regularization terms, ω_0, ω_1, ω_2, and ω_3 for MS-SVDD become equivalent to the regularization terms proposed for S-SVDD when $M = 1$. Hence, MS-SVDD is a more generalized form of S-SVDD, which can form a data description by considering data from multiple modalities.

We update $\mathbf{Q}_m$ by using the gradient of L in (11) with respect to $\mathbf{Q}_m$,

$$\mathbf{Q}_m \leftarrow \mathbf{Q}_m - \eta \Delta L, \tag{20}$$

where η is the learning rate parameter and the gradient of L is calculated as

$$\frac{\partial L}{\partial \mathbf{Q}_m} = 2\sum_{i=1}^{N} \alpha_{m,i} \mathbf{Q}_m \mathbf{x}_{m,i} \mathbf{x}_{m,i}^T - 2\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{n=1}^{M} \mathbf{Q}_n \mathbf{x}_{n,j} \mathbf{x}_{m,i}^T \alpha_{m,i} \alpha_{n,j} + \beta \Delta\omega, \tag{21}$$

where $\Delta\omega$ is the derivative of the regularization term with respect to $\mathbf{Q}_m$

$$\Delta\omega_0 = 0, \tag{22}$$

$$\Delta\omega_1 = 2\mathbf{Q}_m \mathbf{X}_m \mathbf{X}_m^T, \tag{23}$$

$$\Delta\omega_2 = 2\mathbf{Q}_m \mathbf{X}_m \boldsymbol{\alpha}_m \boldsymbol{\alpha}_m^T \mathbf{X}_m^T, \tag{24}$$

$$\Delta\omega_3 = 2\mathbf{Q}_m \mathbf{X}_m \boldsymbol{\lambda}_m \boldsymbol{\lambda}_m^T \mathbf{X}_m^T, \tag{25}$$

$$\Delta\omega_4 = 2\sum_{n=1}^{M} (\mathbf{Q}_n \mathbf{X}_n \mathbf{X}_m^T), \tag{26}$$

$$\Delta\omega_5 = 2\sum_{n=1}^{M} (\mathbf{Q}_n \mathbf{X}_n \boldsymbol{\alpha}_n \boldsymbol{\alpha}_m^T \mathbf{X}_m^T), \tag{27}$$

$$\Delta\omega_6 = 2\sum_{n=1}^{M} (\mathbf{Q}_n \mathbf{X}_n \boldsymbol{\lambda}_n \boldsymbol{\lambda}_m^T \mathbf{X}_m^T). \tag{28}$$

We initialize the $\mathbf{Q}_m$ using PCA. At every iteration, the projection matrix is orthogonalized and normalized so that

$$\mathbf{Q}_m \mathbf{Q}_m^T = \mathbf{I}, \tag{29}$$

where $\mathbf{I}$ is an identity matrix. We use QR decomposition for orthogonalizing and normalizing the projection matrix $\mathbf{Q}_m$. Algorithm 1 describes the overall MS-SVDD algorithm.

Algorithm 1: MS-SVDD optimization.

Inputs : $\mathbf{Z}_m$ for each $m = 1, \ldots, M$, // Input data from all modalities
β, // Regularization parameter for controlling significance of ω
η, // Learning rate parameter
d, // Dimensionality of joint subspace
C, // Regularization parameter in SVDD
M // Total number of modalities

Outputs: $\mathbf{S}_m$ for each $m = 1, \ldots, M$, // Projection matrices for different modalities
R, // Radius of hypersphere
$\boldsymbol{\alpha}$ // Defines the data description

$\mathbf{Z}_m = \mathbf{X}_m$ for linear and NPT case ($\mathbf{K}_m$ for kernel case)
$\mathbf{S}_m = \mathbf{Q}_m$ for linear and NPT case ($\mathbf{W}_m$ for kernel case)

for *m=1:M* **do**
  Initialize $\mathbf{S}_m$ via linear-PCA (kernel-PCA);
end
for *iter* $= 1 :$ max_*iter* **do**
  For each m, map $\mathbf{Z}_m$ to $\mathbf{Y}_m$ using Eq. (2) (Eq. (31));
  Form Y by combining all $\mathbf{Y}_m$'s;
  Solve SVDD in the subspace to obtain $\boldsymbol{\alpha}$ in Eq. (10);
  for *m=1:M* **do**
    Calculate $\boldsymbol{\Delta L}$ using Eq. (21) (Eq. (31)) ;
    Update $\mathbf{S_m} \leftarrow \mathbf{S_m} - \boldsymbol{\eta \Delta L}$;
    Orthogonalize and normalize $\mathbf{S_m}$ using QR decomposition (eigendecomposition);
  end
end
For each m, compute $\mathbf{Y}_m$ using Eq. (2) (Eq. (31));
Form $\mathbf{Y}$ by combining all $\mathbf{Y}_m$'s;
Solve SVDD to obtain the final data description;

3.2. Non-linear MS-SVDD

For non-linear mapping from the original feature spaces to a new shared feature space, we use two approaches. The first approach is based on the standard kernel trick [20] and the second on the Nonlinear Projection Trick (NPT) [37], which is used as a computationally lighter alternative to the kernel trick.

3.2.1. Non-linear MS-SVDD with standard kernel trick

In the non-linear data description, the original data are mapped to a kernel space $\mathcal{F}$ using a non-linear function $\phi(\cdot)$ such that $\mathbf{x}_{m,i} \in \mathbb{R}^{D_m} \rightarrow \phi(\mathbf{x}_{m,i}) \in \mathcal{F}$. The kernel space dimensionality can possibly be infinite. Then the data are projected from the kernel space to $\mathbb{R}^d$ as

$$\mathbf{y}_{m,i} = \mathbf{Q}_m \phi(\mathbf{x}_{m,i}), \quad \forall i \in \{1, \ldots, N\}. \tag{30}$$

In order to calculate $\mathbf{y}_{m,i}$, we use the so-called kernel trick by expressing the projection matrix $\mathbf{Q}_m$ as a linear combination of the training data representations of the respective modality in the kernel space $\mathcal{F}$, leading to

$$\mathbf{y}_{m,i} = \mathbf{W}_m \boldsymbol{\Phi}_m^T \phi(\mathbf{x}_{m,i}) = \mathbf{W}_m \mathbf{k}_{m,i}, \quad \forall i \in \{1, \ldots, N\}, \tag{31}$$

where $\boldsymbol{\Phi}_m \in \mathbb{R}^{|\mathcal{F}| \times N}$ is a matrix formed in $\mathcal{F}$ containing the training data representations of modality m, $\mathbf{W}_m \in \mathbb{R}^{d \times N}$ is a matrix containing the weights for $\boldsymbol{\Phi}_m$ needed to form $\mathbf{Q}_m$, and $\mathbf{k}_{m,i}$ is the ith column of the Gramian matrix, also called as the kernel matrix, $\mathbf{K}_m \in \mathbb{R}^{N \times N}$, having elements equal to $\mathbf{K}_{m,ij} = \phi(\mathbf{x}_{m,i})^T \phi(\mathbf{x}_{m,j})$. In our experiments, we use the Radial Basis Function (RBF) kernel, given by

$$\mathbf{K}_{m,ij} = \exp\left(\frac{-\|\mathbf{x}_{m,i} - \mathbf{x}_{m,j}\|_2^2}{2\sigma^2}\right), \tag{32}$$

where $\sigma > 0$ is a hyperparameter and determines the width of the kernel.

The augmented version of the Lagrangian function now takes the following form:

$$L = \sum_{m=1}^{M}\sum_{i=1}^{N} \alpha_{m,i} \mathbf{k}_{m,i}^T \mathbf{W}_m^T \mathbf{W}_m \mathbf{k}_{m,i} - \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{n=1}^{M}\sum_{j=1}^{N} \alpha_{m,i} \mathbf{k}_{m,i}^T \mathbf{W}_m^T \mathbf{W}_n \mathbf{k}_{n,j} \alpha_{n,j} + \beta\omega. \tag{33}$$

The α's are calculated optimizing (10) with $\mathbf{W}_m$'s fixed, i.e., applying SVDD in the subspace. In the second step, the α's are fixed and $\mathbf{W}_m$'s are updated with the gradient descent:

$$\mathbf{W}_m \leftarrow \mathbf{W}_m - \eta \Delta L, \tag{34}$$

where the gradient is calculated as

$$\frac{\partial L}{\partial \mathbf{W}_m} = 2\sum_{i=1}^{N} \alpha_{m,i}\mathbf{W}_m\mathbf{k}_{m,i}\mathbf{k}_{m,i}^T - 2\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{n=1}^{M}\mathbf{W}_n\mathbf{k}_{n,j}\mathbf{k}_{m,i}^T\alpha_{m,i}\alpha_{n,j} + \beta\Delta\omega. \tag{35}$$

The gradient of the regularization term, $\Delta\omega$, now takes the following forms:

$$\Delta\omega_0 = 0, \tag{36}$$

$$\Delta\omega_1 = 2\mathbf{W}_m\mathbf{K}_m\mathbf{K}_m^T, \tag{37}$$

$$\Delta\omega_2 = 2\mathbf{W}_m\mathbf{K}_m\boldsymbol{\alpha}_m\boldsymbol{\alpha}_m^T\mathbf{K}_m^T, \tag{38}$$

$$\Delta\omega_3 = 2\mathbf{W}_m\mathbf{K}_m\boldsymbol{\lambda}_m\boldsymbol{\lambda}_m^T\mathbf{K}_m^T, \tag{39}$$

$$\Delta\omega_4 = 2\sum_{n=1}^{M}(\mathbf{W}_n\mathbf{K}_n\mathbf{K}_m^T), \tag{40}$$

$$\Delta\omega_5 = 2\sum_{n=1}^{M}(\mathbf{W}_n\mathbf{K}_n\boldsymbol{\alpha}_n\boldsymbol{\alpha}_m^T\mathbf{K}_m^T), \tag{41}$$

$$\Delta\omega_6 = 2\sum_{n=1}^{M}(\mathbf{W}_n\mathbf{K}_n\boldsymbol{\lambda}_n\boldsymbol{\lambda}_m^T\mathbf{K}_m^T). \tag{42}$$

We initialize the matrix $\mathbf{W}_m$ for each mode using kernel-PCA. We orthogonalize and normalize $\mathbf{W}_m$ at every iteration so that

$$\mathbf{W}_m\boldsymbol{\Phi}_m^T\boldsymbol{\Phi}_m\mathbf{W}_m^T = \mathbf{I}. \tag{43}$$

We decompose (43) using eigendecomposition as

$$\mathbf{W}_m\boldsymbol{\Phi}_m^T\boldsymbol{\Phi}_m\mathbf{W}_m^T = \mathbf{V}_m\boldsymbol{\Lambda}_m\mathbf{V}_m^T, \tag{44}$$

where $\boldsymbol{\Phi}_m^T\boldsymbol{\Phi}_m$ is $\mathbf{K}_m$, $\boldsymbol{\Lambda}_m$ is a diagonal matrix containing the eigenvalues of $\mathbf{W}_m\boldsymbol{\Phi}_m^T\boldsymbol{\Phi}_m\mathbf{W}_m^T$ and $\mathbf{V}_m$ contains the corresponding eigenvectors. After further simplification, the normalized projection matrix $\hat{\mathbf{W}}_m$ can be computed as

$$\hat{\mathbf{W}}_m = (\boldsymbol{\Lambda}_m^{\frac{1}{2}})^{+}\mathbf{V}_m^T\mathbf{W}_m, \tag{45}$$

where the + sign denotes pseudo-inverse. For notation simplicity, we set $\mathbf{W}_m = \hat{\mathbf{W}}_m$.

3.2.2. Non-linear MS-SVDD with nonlinear projection trick

The non-linear MS-SVDD using the kernel trick requires computing the eigendecomposition (44) at every iteration. This is computationally expensive and, therefore, we propose an alternative non-linear approach using NPT [37]. Here, a non-linear mapping is applied only at the beginning of the process, while the optimization follows the linear MS-SVDD. In the NPT-based MS-SVDD, we first compute kernel matrix $\mathbf{K}_m$ using (32). In the next step, the computed kernel matrix is centralized as

$$\hat{\mathbf{K}}_m = (\mathbf{I} - \mathbf{E}_N)\mathbf{K}_m(\mathbf{I} - \mathbf{E}_N) \tag{46}$$

where $\hat{\mathbf{K}}_m$ is the centralized kernel matrix and $\mathbf{E}_N$ is $N \times N$ matrix defined as

$$\mathbf{E}_N = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T. \tag{47}$$

$\mathbf{1}_N \in \mathbb{R}^N$is a vector with each element having value of 1. The centralized matrix $\hat{\mathbf{K}}_m$ is decomposed by using eigendecomposition,

$$\hat{\mathbf{K}}_m = \mathbf{U}_m\mathbf{A}_m\mathbf{U}_m^T, \tag{48}$$

where $\mathbf{A}_m$ contains the non-negative eigenvalues of the centered kernel matrix and $\mathbf{U}_m$ contains the corresponding eigenvectors. The data in the reduced dimensional kernel space is obtained as

$$\boldsymbol{\Phi}_m = (\mathbf{A}_m^{\frac{1}{2}})^{+}\mathbf{U}_m^{+}\hat{\mathbf{K}}_m \tag{49}$$

Since we consider NPT as a pure preprocessing step, we continue by considering $\boldsymbol{\Phi}_m$ as our input data, i.e., we set $\mathbf{X}_m = \boldsymbol{\Phi}_m$. Then we follow the linear MS-SVDD. Note that in cases where the number of training samples is high, this pre-processing step can be highly accelerated by following approximations, like the Nyström-based Approximate Kernel Subspace Learning method in [38].

3.3. Test phase

During the test phase, an instance $\mathbf{x}_{m*} \in \mathbb{R}^{D_m}$ (the $*$ in subscript denotes test instance) coming from modality m is projected to the common d-dimensional subspace using (2) for the linear case. For kernel case, first, the kernel vector is computed as

$$\mathbf{k}_{m*} = \boldsymbol{\Phi}_m^T\phi(\mathbf{x}_{m*}) \tag{50}$$

and then projected to the common d-dimensional subspace using (31). For NPT, first the kernel vector $\mathbf{k}_{m^*}$ is computed and then centralized as

$$\hat{\mathbf{k}}_{m*} = (\mathbf{I} - \mathbf{E}_N)[\mathbf{k}_{m*} - \frac{1}{N}\mathbf{K}_m\mathbf{1}_N]. \tag{51}$$

The centralized kernel vector is mapped to

$$\phi_{m*} = (\boldsymbol{\Phi}_m^T)^{+}\hat{\mathbf{k}}_{m*} \tag{52}$$

and then to d-dimensional subspace using (2) (for notation simplicity $\boldsymbol{\phi}_{m^*}$ is considered as $\mathbf{x}_{m^*}$).

The decision to classify the test instance $\mathbf{y}_{m^*}$ as positive or negative is taken on the basis of its distance from the center of hypersphere, i.e.,

$$\|\mathbf{y}_{m*} - \mathbf{a}\|_2^2 = \mathbf{y}_{m*}^T\mathbf{y}_{m*} - 2\sum_{k=1}^{M}\sum_{i=1}^{N}\alpha_{k,i}\mathbf{y}_{m*}^T\mathbf{y}_{k,i} + \sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{n=1}^{M}\sum_{j=1}^{N}\alpha_{k,i}\alpha_{n,j}\mathbf{y}_{k,i}^T\mathbf{y}_{n,j}. \tag{53}$$

The representation $\mathbf{y}_{m^*}$ is assigned to the positive class when $\|\mathbf{y}_{m*} - \mathbf{a}\|_2^2 \leq R^2$ and to the negative class if $\|\mathbf{y}_{m*} - \mathbf{a}\|_2^2 > R^2$, where R^2 is the distance from center $\mathbf{a}$ to any support vector on the boundary,

$$R^2 = \mathbf{v}^T\mathbf{v} - 2\sum_{m=1}^{M}\sum_{i=1}^{N}\alpha_{m,i}\mathbf{y}_{m,i}^T\mathbf{v} + \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{n=1}^{M}\sum_{j=1}^{N}\alpha_{m,i}\alpha_{n,j}\mathbf{y}_{m,i}^T\mathbf{y}_{n,j}, \tag{54}$$

where $\mathbf{v}$ is any support vector in the training set with corresponding α having value $0 < \alpha < C$. Since the items are represented by M different modalities, the final decision for assigning the item to a particular class (either positive or negative) can be taken using different strategies explained in Section 4.3.

3.4. Complexity analysis

The linear version of the proposed method has the following main steps: 1) Initializing the projection matrices via PCA, 2) mapping data from all modalities to a lower d-dimensional shared space, 3) SVDD for obtaining the α values and final data description for all data points coming from M different modalities, 4)

computing the gradient (ΔL) for each modality, 5) updating the projection matrices and 6) QR decomposition for orthogonalizing and normalizing the projection matrices. We analyze each of these steps and then compute the overall complexity of the algorithm:

1. PCA of a matrix is computed by the eigenvalue decomposition of its covariance matrix, so it involves two steps, i.e., computing the covariance matrix and then the eigenvalue decomposition of the obtained covariance matrix. The complexity of calculating covariance matrix and corresponding eigenvalue decomposition for a single modality is $\mathcal{O}\big(ND_m \times \min(N, D_m)\big)$ and $\mathcal{O}\big(D_m^3\big)$, respectively [39]. The complexity of computing PCA for all modalities is $\mathcal{O}\big(\min(N^2D_1, D_1^2N) + D_1^3) + (\min(N^2D_2, D_2^2N) + D_2^3) + \cdots + (\min(N^2D_M, D_M^2N) + D_M^3\big)$. We denote the sum of dimensions of all modalities as $\Sigma_{\mathcal{D}} = D_1 + D_2 + \cdots + D_M$ and similarly the sum of squared dimensions as $\Sigma_{\mathcal{D}^2} = D_1^2 + D_2^2 + \cdots + D_M^2$ (note that $\Sigma_{\mathcal{D}^2} \neq (\Sigma_{\mathcal{D}})^2$) and sum of cubed dimensions as $\Sigma_{\mathcal{D}^3} = D_1^3 + D_2^3 + \cdots + D_M^3$. Hence, the complexity of initializing the projection matrices via PCA becomes $\mathcal{O}\big(\min(N^2\Sigma_{\mathcal{D}}, \Sigma_{\mathcal{D}^2}N) + \Sigma_{\mathcal{D}^3}\big)$.
2. The complexity of mapping data from the original D_m dimensional space to a lower d-dimensional space is the complexity of multiplying $d \times D_m$ and $D_m \times N$, which has the complexity of $\mathcal{O}\big(dD_mN\big)$. Repeating this for all modalities we get $\mathcal{O}\big(d\Sigma_{\mathcal{D}}N\big)$
3. The complexity of SVDD for N data points is $\mathcal{O}\big(N^3\big)$ [40]. For all data points coming from M different modalities it becomes $\mathcal{O}\big(M^3N^3\big)$.
4. The gradient ΔL to update $\mathbf{Q}_m$ is computed using (21), where the second term has the highest complexity (equally high as regularization terms 4–6). Its complexity is $\mathcal{O}(2dN^2D_m\Sigma_{\mathcal{D}})$. As this step is repeated for all modalities the total complexity becomes $\mathcal{O}(2dN^2{\Sigma_{\mathcal{D}}}^2)$.
5. Updating the projection matrices has $\mathcal{O}\big(d\Sigma_{\mathcal{D}}\big)$ complexity.
6. The complexity of QR decomposition for a single modality is $\mathcal{O}(d{D_m}^2)$ [41]. Thus, the overall complexity of QR decompositions for all the modalities is $\mathcal{O}(d\Sigma_{\mathcal{D}^2})$.

Dropping the relatively lower intensive computational steps and adding the rest, the full complexity of the proposed method reduces to $\mathcal{O}\big(\min(N^2\Sigma_{\mathcal{D}}, \Sigma_{\mathcal{D}^2}N) + \Sigma_{\mathcal{D}^3} + M^3N^3\big)$. Assuming that the total number of samples M^*N is always greater than $\mathcal{D}$ and $M < < N$, the time complexity of (a single iteration of) our proposed algorithm in terms of the big $\mathcal{O}$ notation is $\mathcal{O}(N^3)$. In the testing phase, each representation of a test sample in each modality is projected to the d-dimensional subspace and then its distance is compared to R. This has the total complexity of $\mathcal{O}(d\Sigma_{\mathcal{D}} + Md)$.

For the non-linear version with NPT, the kernel matrix $\mathbf{K}_m$ is first formed which has the complexity of $\mathcal{O}(D_mN^2)$. Then the kernel matrix is centralized and decomposed by using eigendecomposition. Both of these steps have the complexity of $\mathcal{O}(N^3)$. As the data dimensionality in the remaining steps of the proposed method changes from D_m to N, the total complexity of the remaining steps becomes $\mathcal{O}\big(MN^3 + M^3N^3\big)$. Thus, the overall complexity in terms of the big $\mathcal{O}$ notation remains at $\mathcal{O}\big(N^3\big)$ for $M < < N$, while in practice the computational complexity is higher (by a scalar multiplier c) than for the linear version. Also for the non-linear version with the standard kernel trick, the overall complexity remains the same, but the kernel mapping is repeated at every iteration and, thus, the scalar c becomes larger for the overall training process. The testing complexity of the non-linear methods increases to $\mathcal{O}(N\Sigma_{\mathcal{D}} + dMN + Md)$.

4. Experiments

4.1. Datasets and prepossessing

To evaluate the proposed method, we performed different sets of experiments over 5 datasets. Robot Execution Failures dataset, Single Proton Emission Computed Tomography (SPECTF) heart dataset, and Ionosphere dataset were downloaded from UC Irvine (UCI) machine learning repository [34]. Caltech-7 dataset and Handwritten dataset were downloaded from a repository for multi-view learning [42]. The details of datasets and experiments are as follows.

The first set of experiments was performed on the Robot Execution Failures dataset [43]. In Robot Execution Failures dataset, force and torque measurements are collected at regular intervals of time after a task failure is detected. The dataset is divided into five different learning problems (LP) corresponding to different triggering events:

- **LP1:** Failures in approach to grasp position
- **LP2:** Failures in the transfer of a part
- **LP3:** Position of the part after a transfer failure
- **LP4:** Failures in approach to ungrasp position
- **LP5:** Failures in motion with part

The total number of instances and the distribution of the classes are given in Table 1. All instances are given as 15 samples collected at 315 ms regular time intervals for each sensor. For this dataset, we consider all the instances belonging to the normal class as the target class and the remaining classes as the non-target data. Hence, we have two modalities (torque and force measurements), and we consider the dataset as a one-class classification problem.

The second set of experiments was performed SPECTF heart dataset [44]. The SPECTF heart dataset consists of two sets of features corresponding to rest and stress condition SPECTF images of different subjects. The training set consists of 40 examples diagnosed as healthy heart muscle perfusions and 40 diagnosed as pathological perfusions. The test set consists of 15 instances of healthy heart muscle perfusions and 172 from instances diagnosed as pathological perfusions. We convert this to a multimodal one-class classification problem by considering the rest and stress conditions as different modalities and by selecting the healthy heart muscle perfusions as our target class.

The third set of experiments was performed over the Caltech-7 dataset. We used Gabor feature and Wavelet moments as our two different modalities. The dataset contains 1474 total samples from 7 different classes. We selected faces (435 samples) as our target class and the rest of the classes all together (1039 samples) as the outlier class.

We used Ionosphere dataset for the fourth set of experiments. The categories in this dataset are described by two attributes per pulse number resulting from the complex electromagnetic signal, processed by an autocorrelation function. We used the two attributes (real and complex) for each pulse as two different modalities and the attribute “good” as our target class. The total number of samples in this dataset is 351, out of which 225 are from the target class (good), and the rest of 126 samples are from outlier class (bad).

For the fifth set of experiments, we used Handwritten dataset. We considered the samples of numeral 0 as the target. In the Handwritten dataset, the total number of samples is 2000, out of which 200 are from the target class. The rest of the 1800 samples are considered as an outlier class. We used the Zernike moment (ZER) and morphological (MOR) features as our two different modalities.

Table 1
Robot execution failures dataset.

Learning problem	Instances	Classes and distribution
LP1	88	24% normal 19% collision 18% front collision 39% obstruction
LP2	47	43% normal 13% front collision 15% back collision 11% collision to the right 19% collision to the left
LP3	47	43% ok 19% slightly moved 32% moved 06% lost
LP4	117	21% normal 62% collision 18% obstruction
LP5	164	27% normal 16% bottom collision 13% bottom obstruction 29% collision in part 16% collision in tool

4.2. Experimental setup

For the Robot Execution Failures dataset, Ionosphere dataset, Caltech-7 dataset, and Handwritten dataset, we performed our experiments on 70-30% split for training and testing sets. We selected the 70-30% split randomly 5 times, keeping the distribution of classes similar to the original data. To tune the hyperparameters for final testing, we did 5-fold cross-validation on the training set, where the (70%) training data are divided into 5 different sets, and each time one set is used for validation while all the others for training. The process was repeated 5 times until all the sets have been used as validation sets. For SPECTF heart dataset, the train and test sets are given with the dataset. We did 5-fold cross-validation on the training set to optimize the hyperparameters.

For all datasets, the models were trained by using samples from the positive class only, while testing was carried out using all the classes. The hyperparameters were selected from the following ranges:

- $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}\}$,
- $C \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$,
- $\sigma \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$,
- $d \in \{1, 2, 3, 4, 5, 10, 20, 50, 100\}$,
- $\eta = 0.1$.

Here, we restricted the dimension d of the shared subspace as $d < \min\{D_1, \ldots, D_M\}$ for a given dataset, where D_m is the dimensionality of modality m. For competing methods, the features from different modalities were concatenated before training the model. We also report the results of the competing methods by considering data from one modality at a time for training and testing. For competing methods, the hyperparameters were selected from the same ranges as mentioned above.

4.3. Decision strategies

During testing, after the common compact representation of all modalities was formed, each representation (modality) of an instance was mapped to the lower-dimensional subspace via corresponding projection matrix and classified as described in Section 3.3. The following four strategies were used to decide the final class for the instance:

- **Decision strategy 1** (also called the AND gate)**:** The test instance is assigned the target label if the representations from all modalities for that particular instance are classified to the target class and the non-target label otherwise.
- **Decision strategy 2** (also called as the OR gate)**:** The final decision is taken on the basis of the OR gate principle, i.e., if a representation of an instance from any of the modalities is classified to the target class, the overall decision for that particular instance is taken in favor of the target class.
- **Decision strategy 3:** The final classification decision is made on the basis of first modality, i.e., if the representation from the first modality is assigned to a particular class, the overall classification is made following that.
- **Decision strategy 4:** The overall decision is taken on the basis of the label assigned to the representation from the second modality.

It should be noted that for more than two modalities, different decision strategies, such as majority vote, might be more suitable.

4.4. Evaluation criteria

One-class classification models can be evaluated using different metrics. These metrics are decided on the basis of the goals of a given application. For example, in outlier detection, the focus is on detecting negative instances accurately. The most common metrics in one-class classification are true positive rate (tpr), and true negative rate (tnr). The former, also called as recall, sensitivity, or hit rate, is the proportion of positive instances that is classified by the trained model as positive correctly:

$$tpr = \frac{tp}{p}, \tag{55}$$

where tp is the number of positive samples classified correctly and p is the total number of positive samples in the test set. The latter, tnr, also called as specificity, is defined as

$$tnr = \frac{tn}{n}, \tag{56}$$

where tn is the number of negative samples classified correctly and n is the total number of negative samples in the test set. Accuracy ($accu$) is measured as the ratio of the number of correctly classified instances to the total number of instances:

$$accu = \frac{tp + tn}{p + n}. \tag{57}$$

Precision (pre) measures the proportion of instances classified positive which really are positive:

$$pre = \frac{tp}{tp + fp}, \tag{58}$$

where fp is the number of false positives. Another useful measure is $F1$ measure, which is the harmonic mean of pre and tpr:

$$F1 = 2 \times \frac{pre \times tpr}{pre + tpr}. \tag{59}$$

Geometric mean (gm) is defined as the square root of the product of sensitivity and specificity:

$$gm = \sqrt{tpr \times tnr}. \tag{60}$$

gm has been used by many researchers for imbalanced datasets. Since it takes into consideration both sensitivity and specificity, we opted to finetune hyperparameters based on the gm score on the validation data.

4.5. Experimental results and discussion

InTables 2–5, we report the average of different evaluation metrics over the five data splits for Robot Execution Failures dataset, Caltech-7 dataset, Ionosphere dataset, and Handwritten dataset, respectively, for both linear and non-linear versions of the applied

Table 2
Test results for robot execution failures dataset.

	Linear						Non-linear					
	accu	tpr	tnr	pre	F1	gm	accu	tpr	tnr	pre	F1	gm
Proposed method												
MS-SVDD$\omega_2 ds3$	0.97	0.97	0.97	0.93	0.95	**0.97**	0.94	0.98	0.92	0.83	0.90	0.95
MS-SVDD$\omega_5 ds3$	0.97	0.95	0.97	0.93	0.94	0.96	0.94	0.98	0.92	0.83	0.90	0.95
Concatenated features												
S-SVDDψ_1	0.66	0.89	0.57	0.46	0.60	0.71	0.94	0.84	0.98	0.95	0.89	0.91
S-SVDDψ_2	0.70	0.80	0.66	0.58	0.60	0.70	0.92	0.90	0.93	0.84	0.87	0.91
S-SVDDψ_3	0.66	0.78	0.61	0.46	0.56	0.67	0.93	0.93	0.93	0.85	0.89	0.93
S-SVDDψ_4	0.64	0.94	0.52	0.44	0.60	0.70	0.96	0.90	0.98	0.96	0.93	0.94
OC-SVM	0.51	0.47	0.52	0.28	0.35	0.49	0.86	0.49	1.00	1.00	0.65	0.70
SVDD	0.97	0.91	0.99	0.98	0.95	0.95	0.95	0.85	0.99	0.98	0.91	0.92
Force measurements												
S-SVDDψ_1	0.76	0.88	0.71	0.55	0.67	0.79	0.96	0.90	0.98	0.95	0.92	0.94
S-SVDDψ_2	0.77	0.94	0.71	0.56	0.70	0.82	0.96	0.90	0.98	0.95	0.92	0.94
S-SVDDψ_3	0.73	0.70	0.74	0.51	0.58	0.71	0.96	0.91	0.98	0.95	0.93	0.94
S-SVDDψ_4	0.76	0.85	0.72	0.54	0.66	0.78	0.93	0.82	0.98	0.95	0.84	0.88
OC-SVM	0.50	0.53	0.49	0.29	0.37	0.51	0.86	0.49	1.00	1.00	0.65	0.70
SVDD	0.97	0.90	0.99	0.98	0.94	0.95	0.97	0.92	0.99	0.98	0.95	**0.96**
Torque measurements												
S-SVDDψ_1	0.59	0.96	0.44	0.41	0.57	0.65	0.97	0.89	1.00	1.00	0.94	0.94
S-SVDDψ_2	0.61	0.94	0.48	0.42	0.57	0.67	0.71	0.66	0.73	0.51	0.54	0.51
S-SVDDψ_3	0.62	0.92	0.50	0.43	0.58	0.67	0.92	0.76	0.99	0.97	0.82	0.85
S-SVDDψ_4	0.61	0.96	0.48	0.42	0.58	0.68	0.76	0.76	0.76	0.76	0.71	0.66
OC-SVM	0.52	0.59	0.49	0.31	0.40	0.53	0.84	0.58	0.94	0.81	0.66	0.73
SVDD	0.90	0.95	0.88	0.76	0.84	0.91	0.91	0.88	0.92	0.81	0.84	0.90

Table 3
Test results for Caltech-7 dataset.

	Linear						Non-linear					
	accu	tpr	tnr	pre	F1	gm	accu	tpr	tnr	pre	F1	gm
Proposed method												
MS-SVDD$\omega_1 ds1$	0.91	0.96	0.89	0.78	0.86	0.92	0.94	0.98	0.92	0.85	0.91	**0.95**
MS-SVDD$\omega_4 ds1$	0.91	0.95	0.89	0.78	0.86	0.92	0.94	0.95	0.94	0.88	0.91	**0.95**
Concatenated features												
S-SVDDψ_1	0.65	0.96	0.52	0.46	0.62	0.71	0.37	0.35	0.38	0.15	0.20	0.23
S-SVDDψ_2	0.67	0.92	0.57	0.48	0.63	0.72	0.66	0.69	0.64	0.39	0.48	0.53
S-SVDDψ_3	0.71	0.84	0.66	0.59	0.65	0.69	0.90	0.79	0.94	0.86	0.81	0.86
S-SVDDψ_4	0.62	0.96	0.47	0.46	0.61	0.66	0.87	0.61	0.97	0.91	0.72	0.76
OC-SVM	0.22	0.47	0.12	0.18	0.26	0.22	0.86	0.53	1.00	0.99	0.69	0.73
SVDD	0.92	0.94	0.91	0.81	0.87	**0.93**	0.96	0.94	0.97	0.93	0.94	**0.95**
Gabor feature												
S-SVDDψ_1	0.68	0.72	0.67	0.47	0.57	0.69	0.46	0.84	0.31	0.33	0.48	0.50
S-SVDDψ_2	0.68	0.72	0.67	0.47	0.57	0.69	0.54	0.78	0.44	0.46	0.52	0.50
S-SVDDψ_3	0.61	0.74	0.55	0.45	0.52	0.58	0.76	0.68	0.80	0.65	0.63	0.71
S-SVDDψ_4	0.70	0.74	0.68	0.49	0.59	0.71	0.78	0.39	0.94	0.80	0.46	0.55
OC-SVM	0.43	0.53	0.40	0.27	0.36	0.45	0.79	0.55	0.89	0.69	0.61	0.70
SVDD	0.76	0.70	0.78	0.57	0.63	0.74	0.74	0.92	0.67	0.55	0.68	0.78
Wavelet moments												
S-SVDDψ_1	0.70	0.73	0.68	0.50	0.59	0.69	0.54	0.44	0.58	0.22	0.26	0.24
S-SVDDψ_2	0.71	0.73	0.70	0.52	0.60	0.70	0.51	0.93	0.33	0.41	0.55	0.42
S-SVDDψ_3	0.50	0.93	0.33	0.38	0.54	0.50	0.79	0.38	0.96	0.65	0.44	0.51
S-SVDDψ_4	0.56	0.88	0.42	0.40	0.54	0.59	0.61	0.51	0.65	0.50	0.36	0.30
OC-SVM	0.21	0.48	0.10	0.18	0.26	0.21	0.84	0.48	0.99	0.97	0.64	0.69
SVDD	0.91	0.94	0.89	0.79	0.85	0.91	0.94	0.97	0.93	0.85	0.91	**0.95**

methods. In Table 6, we report the results on the test set for the SPECTF heart dataset. In these tables, we only show the best performing versions of the proposed method, along with all competing methods. We compare our results with OC-SVM [26], SVDD [25], and S-SVDD [36]. In S-SVDD, different regularization terms (ψ's) were proposed and, hence, we compare MS-SVDD with all proposed regularization terms of S-SVDD. We use kernel version of the competing methods for non-linear comparisons. In these tables, we report the best performing non-linear version of MS-SVDD for corresponding datasets. To analyze the different regularization terms and decision strategies for the proposed method, we also report the exhaustive results obtained by different settings in the supplementary material in Tables 1–5. The best results in terms of *gm* are reported as in bold formatting.

For the Robot Execution Failures dataset (Table 2), our proposed method outperforms all the competing methods in the linear case. The results achieved by the linear version of the proposed MS-SVDD method are overall best also compared to the non-linear methods. Table 2 shows that using decision strategy 3 with constraint ω_2 (all support vectors and outliers from the corresponding modality considered for the update of the corresponding $\mathbf{Q}_m$) yields the best overall results for the robot dataset. In the

Table 4
Test results for Ionosphere dataset.

	Linear						Non-linear					
	accu	tpr	tnr	pre	F1	gm	accu	tpr	tnr	pre	F1	gm
Proposed method												
MS-SVDD$\omega_2 ds4$	0.87	0.95	0.73	0.87	0.91	0.83	0.76	0.86	0.59	0.79	0.82	0.71
MS-SVDD$\omega_1 ds4$	0.83	0.91	0.69	0.84	0.87	0.79	0.88	0.95	0.74	0.87	0.91	0.84
Concatenated features												
S-SVDDψ_1	0.69	0.88	0.32	0.69	0.77	0.53	0.49	0.37	0.69	0.55	0.39	0.28
S-SVDDψ_2	0.69	0.89	0.31	0.69	0.77	0.51	0.74	0.60	0.98	0.98	0.75	0.77
S-SVDDψ_3	0.58	0.63	0.48	0.66	0.62	0.51	0.72	0.77	0.62	0.83	0.77	0.63
S-SVDDψ_4	0.72	0.98	0.23	0.70	0.82	0.43	0.66	0.61	0.77	0.88	0.67	0.62
OC-SVM	0.38	0.39	0.34	0.52	0.45	0.37	0.66	0.48	0.97	0.97	0.63	0.67
SVDD	0.87	0.93	0.76	0.88	0.90	**0.84**	0.89	0.94	0.78	0.89	0.92	0.86
Real												
S-SVDDψ_1	0.81	0.99	0.50	0.78	0.87	0.69	0.54	0.36	0.86	0.67	0.43	0.46
S-SVDDψ_2	0.80	0.99	0.47	0.78	0.87	0.67	0.62	0.49	0.86	0.87	0.61	0.64
S-SVDDψ_3	0.81	0.99	0.49	0.78	0.87	0.68	0.68	0.63	0.78	0.86	0.70	0.68
S-SVDDψ_4	0.81	0.99	0.50	0.78	0.87	0.70	0.58	0.45	0.83	0.85	0.53	0.56
OC-SVM	0.49	0.52	0.42	0.61	0.56	0.46	0.68	0.56	0.89	0.93	0.67	0.69
SVDD	0.88	0.95	0.74	0.87	0.91	**0.84**	0.89	0.94	0.81	0.90	0.92	**0.87**
Complex												
S-SVDDψ_1	0.50	0.37	0.72	0.70	0.49	0.51	0.43	0.27	0.71	0.52	0.30	0.34
S-SVDDψ_2	0.47	0.35	0.69	0.67	0.46	0.49	0.66	0.56	0.83	0.85	0.68	0.68
S-SVDDψ_3	0.53	0.57	0.46	0.67	0.58	0.39	0.65	0.65	0.65	0.78	0.70	0.63
S-SVDDψ_4	0.50	0.38	0.72	0.70	0.49	0.52	0.63	0.64	0.62	0.76	0.69	0.62
OC-SVM	0.40	0.31	0.57	0.56	0.40	0.42	0.66	0.59	0.78	0.84	0.69	0.67
SVDD	0.77	0.89	0.55	0.79	0.83	0.70	0.79	0.91	0.58	0.80	0.85	0.72

Table 5
Test results for Handwritten dataset.

	Linear						Non-linear					
	accu	tpr	tnr	pre	F1	gm	accu	tpr	tnr	pre	F1	gm
Proposed method												
MS-SVDD$\omega_4 ds4$	0.98	0.99	0.98	0.90	0.93	**0.98**	0.99	0.99	1.00	0.98	0.98	**0.99**
MS-SVDD$\omega_4 ds1$	0.98	0.90	0.99	0.89	0.89	0.94	0.98	0.95	0.99	0.91	0.93	0.97
Concatenated features												
S-SVDDψ_1	0.78	0.92	0.76	0.34	0.49	0.83	0.53	0.40	0.54	0.05	0.09	0.14
S-SVDDψ_2	0.82	0.88	0.81	0.40	0.54	0.84	0.62	0.66	0.61	0.18	0.25	0.44
S-SVDDψ_3	0.82	0.97	0.81	0.39	0.55	0.88	0.63	0.58	0.64	0.20	0.25	0.30
S-SVDDψ_4	0.84	0.92	0.83	0.42	0.56	0.87	0.71	0.39	0.75	0.08	0.13	0.17
OC-SVM	0.50	0.51	0.50	0.12	0.19	0.49	0.95	0.51	1.00	1.00	0.68	0.71
SVDD	0.95	0.93	0.95	0.69	0.79	0.94	0.95	0.92	0.96	0.74	0.81	0.94
ZER												
S-SVDDψ_1	0.55	0.92	0.51	0.18	0.30	0.68	0.59	0.41	0.61	0.06	0.10	0.24
S-SVDDψ_2	0.52	0.88	0.48	0.17	0.28	0.64	0.62	0.78	0.60	0.17	0.27	0.48
S-SVDDψ_3	0.50	0.96	0.45	0.19	0.31	0.63	0.57	0.61	0.57	0.31	0.20	0.37
S-SVDDψ_4	0.64	0.90	0.61	0.21	0.34	0.74	0.55	0.60	0.54	0.09	0.15	0.24
OC-SVM	0.43	0.42	0.43	0.09	0.14	0.41	0.95	0.52	1.00	0.93	0.67	0.72
SVDD	0.88	0.90	0.88	0.47	0.61	0.89	0.92	0.88	0.92	0.56	0.68	0.90
MOR												
S-SVDDψ_1	0.84	0.99	0.82	0.48	0.61	0.90	0.84	0.01	0.93	0.00	0.00	0.03
S-SVDDψ_2	0.92	0.99	0.91	0.66	0.76	0.95	0.58	0.44	0.60	0.43	0.22	0.20
S-SVDDψ_3	0.86	0.99	0.84	0.52	0.64	0.91	0.61	0.70	0.60	0.44	0.42	0.36
S-SVDDψ_4	0.84	0.99	0.82	0.48	0.61	0.90	0.25	0.67	0.20	0.27	0.14	0.04
OC-SVM	0.54	0.45	0.55	0.13	0.18	0.39	0.99	0.87	1.00	1.00	0.93	0.93
SVDD	0.93	0.91	0.93	0.75	0.78	0.92	0.99	0.96	1.00	1.00	0.98	0.98

non-linear case, the best performance for the proposed method is achieved by using the kernel trick with either constraint type ω_2 or ω_5, both with decision strategy 3.

We also notice that the first modality (force measurements) is vital in taking the final decision as in both linear and non-linear cases, the best results are obtained when the decision is taken based on the first modality (decision strategy 3). The importance of the first modality is also evident from the results of the competing methods as the best results are obtained when using force measurements only. The results on the concatenated features are slightly worse, and the results using the torque measurements are clearly worse. Nevertheless, the proposed multimodal approach has managed to boost the results by combining information from both modalities.

For the Caltech-7 dataset, in the linear case, MS-SVDD performs better than all other methods with a single modality. Overall, only SVDD using concatenated features outperforms MS-SVDD and the margin is small. In the non-linear case, MS-SVDD obtains the best results along with SVDD. In terms of *tpr*, MS-SVDD outperforms all the other methods in the non-linear case while maintaining rea-

Table 6
Test results for SPECTF heart dataset.

	Linear						Non-linear					
	accu	tpr	tnr	pre	F1	gm	accu	tpr	tnr	pre	F1	gm
Proposed method												
MS-SVDD$\omega_0 ds1$	0.78	0.80	0.78	0.24	0.37	**0.79**	0.55	0.60	0.55	0.10	0.18	0.57
MS-SVDD$\omega_2 ds1$	0.78	0.80	0.77	0.24	0.36	**0.79**	0.80	0.73	0.80	0.24	0.37	**0.77**
Concatenated features												
S-SVDDψ_1	0.71	0.53	0.73	0.15	0.23	0.62	0.77	0.60	0.78	0.20	0.30	0.69
S-SVDDψ_2	0.69	0.87	0.67	0.19	0.31	0.76	0.77	0.60	0.78	0.20	0.30	0.69
S-SVDDψ_3	0.66	0.93	0.64	0.18	0.31	0.77	0.77	0.60	0.78	0.20	0.30	0.69
S-SVDDψ_4	0.56	0.67	0.55	0.11	0.19	0.60	0.77	0.60	0.78	0.20	0.30	0.69
OC-SVM	0.86	0.27	0.91	0.20	0.23	0.49	0.76	0.73	0.77	0.22	0.33	0.75
SVDD	0.69	0.73	0.69	0.17	0.28	0.71	0.75	0.67	0.76	0.19	0.30	0.71
Rest Mode												
S-SVDDψ_1	0.50	0.73	0.48	0.11	0.19	0.59	0.46	0.87	0.42	0.12	0.20	0.61
S-SVDDψ_2	0.58	0.87	0.55	0.14	0.25	0.69	0.77	0.53	0.79	0.18	0.27	0.65
S-SVDDψ_3	0.40	0.80	0.37	0.10	0.18	0.54	0.79	0.47	0.81	0.18	0.26	0.62
S-SVDDψ_4	0.38	0.87	0.34	0.10	0.18	0.54	0.60	0.87	0.58	0.15	0.26	0.71
OC-SVM	0.76	0.60	0.77	0.19	0.29	0.68	0.61	0.80	0.60	0.15	0.25	0.69
SVDD	0.59	0.73	0.58	0.13	0.22	0.65	0.59	0.73	0.58	0.13	0.22	0.65
Stress Mode												
S-SVDDψ_1	0.53	0.47	0.53	0.08	0.14	0.50	0.68	0.73	0.67	0.16	0.27	0.70
S-SVDDψ_2	0.65	0.80	0.63	0.16	0.27	0.71	0.75	0.53	0.77	0.17	0.26	0.64
S-SVDDψ_3	0.73	0.67	0.73	0.18	0.28	0.70	0.70	0.73	0.70	0.17	0.28	0.72
S-SVDDψ_4	0.55	0.93	0.52	0.14	0.25	0.69	0.75	0.53	0.77	0.17	0.26	0.64
OC-SVM	0.86	0.20	0.91	0.17	0.18	0.43	0.73	0.60	0.74	0.17	0.26	0.67
SVDD	0.76	0.60	0.77	0.19	0.29	0.68	0.78	0.53	0.80	0.19	0.28	0.65

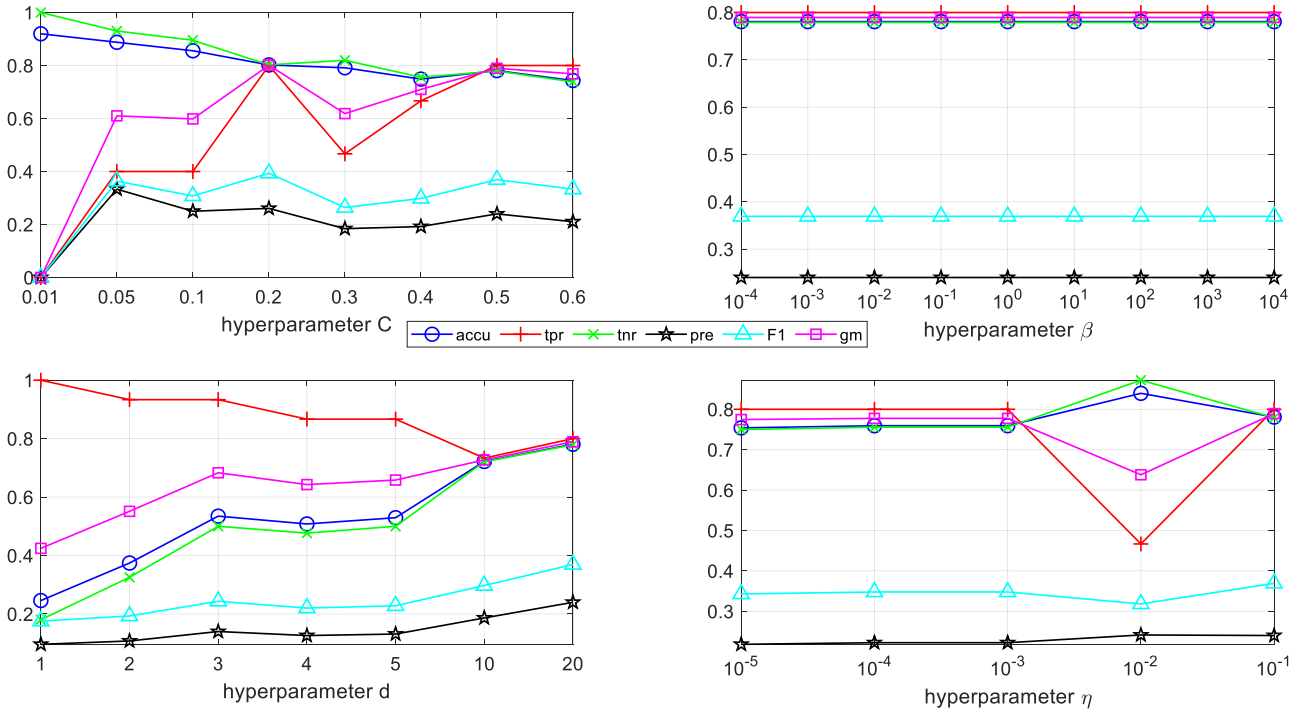

Fig. 2. Hyperparameters sensitivity analysis for $\omega_0 ds1$.

sonably good *tnr*. We also notice that both modalities are vital in taking the final decision as the best performance of MS-SVDD is obtained by decision strategy 1 (AND gate).

For Ionosphere dataset, only SVDD applied on concatenated features or the first modality outperforms MS-SVDD in terms of *gm*. Nevertheless, the performance of MS-SVDD is competitive as shown also by the top results obtained by the other performance metrics such as *F1* measure. In case of MS-SVDD, the second modality (Complex) is found to be more vital for taking the final decision.

For the Handwritten dataset, MS-SVDD outperforms all competing methods in both linear and non-linear cases. It is noticed that decision strategy 4 yields the best results in both linear and non-linear cases for MS-SVDD, i.e., MOR features are more vital than ZER features.

For SPECTF heart dataset, in both linear and non-linear cases, the best results are achieved by MS-SVDD. We note that ω_0 (no constraint used) and ω_2, where all support vectors and outliers are used to describe the class variance for the update of the corresponding $\mathbf{Q}_m$ in decision strategy 1 yield the best overall results.

We compare the results for different variant of MS-SVDD in Tables 1–5 of the supplementary material. Overall in all datasets, NPT is found to be more robust than the kernel version. Linear MS-SVDD is found to perform best over 2 datasets, similar to the NPT version, which performs best on two datasets as well. The kernel MS-SVDD performs best on one out of five datasets as compared to linear and NPT version of MS-SVDD. All the relevant codes (implementation) for the proposed method are available online at [45].

We also carried out a sensitivity analysis of different hyperparameters for linear MS-SVDD over SPECTF heart dataset. To analyze the sensitivity of MS-SVDD for each hyperparameter, we fix the other hyperparameters to their optimal values and record the performance with all the considered hyperparameter values. Fig. 2 shows as an example the results for decision strategy 1 without any constraint. For the other decision strategies and constraints, we show the results in Figures 1–27 in the supplementary material. We note a trend of increase in *tpr* and decrease in *tnr* with the increase of value for hyperparameter C. We also noticed that the performance of trained models are relatively less sensitive to the hyperparameter β as compared to other hyperparameters. For hyperparameter d, initially, there is a noticeable rise in the performance of the trained model; however, after certain value, the change seems to be very small. For hyperparameter η, we notice that precision and *F1* measure stay stable with changing its value.

We also report the numerical training and testing time (in milliseconds) in the supplementary material (Tables 1–10) for all methods over all datasets used in the experiments. In the majority of cases, the proposed method has a higher computational cost than the competing methods, but generally, the difference is in the fractions of a second, which is negligible for datasets used in this work. It is also evident from the numerical results that the time complexity of the proposed method is higher mainly in the training phase, while in the testing phase the difference is negligible. This is as expected based on the complexity analysis in Section 3.4.

5. Conclusion

In this paper, a new multimodal one-class classification method is proposed. The proposed method iteratively transforms data from all the modalities to a new shared subspace optimized for data description in multimodal one-class classification tasks. We derived linear and two different non-linear versions along with a selection of different regularization terms. According to the best of our knowledge, this is the first work in the field of subspace learning for multimodal one-class classification. We conducted experiments comparing the different versions of MS-SVDD and performed comparisons against other one-class classification methods using either concatenated representations or a single modality at a time.

In most cases, linear and NPT version of MS-SVDD outperformed all the competing methods in our experiments. NPT turned out to be more stable than the kernel version. We noticed that the optimal decision strategy depends on the usefulness of different modalities. If a particular modality is more informative than other(s), then it is useful to use that particular modality for making the final decision. Nevertheless, MS-SVDD can improve the results as compared to using a single modality only. If the modalities are more balanced, the AND gate strategy seems to perform better.

MS-SVDD can be interpreted and used in many ways for different one-class multimodal problems. It can be used for anomaly detection and detection of a specific class such as speaker verification and face recognition. In the future, we intend to try different kernels and model-based decision strategies for the proposed method. We also intend to propose changes in the boundary shape (other than spherical) for enclosing the target data in subspace. There is also room for research in other one-class classification techniques for multimodal subspace learning.

Declaration of Competing Interest

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

Acknowledgement

This work was supported by the NSF-Business Finland Center for Visual and Decision Informatics project Co-Botics, jointly sponsored by Tieto Oy Finland and CA Software.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2020.107648.

References

[1] Y. Qu, G. Zhang, Z. Zou, Z. Liu, J. Mao, Active multimodal sensor system for target recognition and tracking, Sensors 17 (7) (2017) 1518.
[2] Y. Zhang, B. Song, X. Du, M. Guizani, Vehicle tracking using surveillance with multimodal data fusion, IEEE Trans. Intell. Transp. Syst. 19 (99) (2018) 1–9.
[3] S. Kye, J. Moon, J. Lee, I. Choi, D. Cheon, K. Lee, Multimodal data collection framework for mental stress monitoring, in: Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers, ACM, 2017, pp. 822–829.
[4] Z. Gu, B. Lang, T. Yue, L. Huang, Learning joint multimodal representation based on multi-fusion deep neural networks, in: International Conference on Neural Information Processing, Springer, 2017, pp. 276–285.
[5] A. Iosifidis, M. Gabbouj, Multi-class support vector machine classifiers using intrinsic and penalty graphs, Pattern Recognit. 55 (2016) 231–246.
[6] S.S. Khan, M.G. Madden, One-class classification: taxonomy of study and review of techniques, Knowl. Eng. Rev. 29 (3) (2014) 345–374.
[7] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, arXiv:1304.5634 (2013).
[8] G. Cao, A. Iosifidis, K. Chen, M. Gabbouj, Generalized multi-view embedding for visual recognition and cross-modal retrieval, IEEE Trans. Cybern. 48 (9) (2018) 2542–2555.
[9] P. Khante, Learning attributes of real-world objects by clustering multimodal sensory data, The University of Texas, 2017 Ph.D. thesis.
[10] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: a survey and taxonomy, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2) (2018) 423–443.
[11] M. Heckmann, Audio-visual word prominence detection from clean and noisy speech, Comput. Speech Lang. 48 (2018) 15–30.
[12] G. Cao, A. Iosifidis, M. Gabbouj, Multi-view nonparametric discriminant analysis for image retrieval and recognition, IEEE Signal Process. Lett. 24 (10) (2017) 1537–1541.
[13] L.-l. Chen, Y. Zhao, P.-f. Ye, J. Zhang, J.-z. Zou, Detecting driving stress in physiological signals based on multimodal feature analysis and kernel classifiers, Expert Syst. Appl. 85 (2017) 279–291.
[14] S. Venugopalan, L.A. Hendricks, M. Rohrbach, R.J. Mooney, T. Darrell, K. Saenko, Captioning images with diverse objects., in: CVPR, vol. 3, 2017, p. 8.
[15] M. Maimaitijiang, V. Sagan, P. Sidike, S. Hartling, F. Esposito, F.B. Fritschi, Soybean yield prediction from UAV using multimodal data fusion and deep learning, Remote Sens. Environ. 237 (2020) 111599.
[16] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: a survey and taxonomy, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2) (2019) 423–443.
[17] H. Hotelling, Relations between two sets of variates, Biometrika 28 (3/4) (1936) 321–377.
[18] J. Benesty, I. Cohen, Canonical correlation analysis, in: Canonical Correlation Analysis in Speech Enhancement, Springer, 2018, pp. 5–14.
[19] P. Xu, Q. Yin, Y. Huang, Y.-Z. Song, Z. Ma, L. Wang, T. Xiang, W.B. Kleijn, J. Guo, Cross-modal subspace learning for fine-grained sketch-based image retrieval, Neurocomputing 278 (2018) 75–86.
[20] B. Scholkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT press, 2001.
[21] M.S. Sadooghi, S.E. Khadem, Improving one class support vector machine novelty detection scheme using nonlinear features, Pattern Recognit. 83 (2018) 14–33.
[22] B. Schölkopf, A. Smola, K.-R. Müller, Kernel principal component analysis, in: International Conference on Artificial Neural Networks, Springer, 1997, pp. 583–588.
[23] T. Kefi-Fatteh, R. Ksantini, M.-B. Kaâniche, A. Bouhoula, A novel incremental one-class support vector machine based on low variance direction, Pattern Recognit. 91 (2019) 308–321.
[24] R. Raghavendra, K.B. Raja, S. Venkatesh, C. Busch, Improved ear verification after surgery-an approach based on collaborative representation of locally competitive features, Pattern Recognit. 83 (2018) 416–429.
[25] D.M. Tax, R.P. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.

[26] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, Sv estimation of a distribution's support, Adv. Neural Inf. Process. Syst. 12 (1999).
[27] R. Sadeghi, J. Hamidzadeh, Automatic support vector data description, Soft Comput. 22 (1) (2018) 147–158.
[28] M. El Boujnouni, M. Jedra, N. Zahid, Support vector domain description with a new confidence coefficient, in: Intelligent Systems: Theories and Applications (SITA-14), 2014 9th International Conference on, IEEE, 2014, pp. 1–8.
[29] Y.-S. Jeong, R. Jayaraman, Support vector-based algorithms with weighted dynamic time warping kernel function for time series classification, Knowl. Based Syst. 75 (2015) 184–191.
[30] Y. Forghani, H.S. Yazdi, S. Effati, R.S. Tabrizi, Support vector data description by using hyper-ellipse instead of hyper-sphere, in: Computer and Knowledge Engineering (ICCKE), 2011 1st International eConference on, IEEE, 2011, pp. 22–27.
[31] V. Mygdalis, A. Iosifidis, A. Tefas, I. Pitas, Graph embedded one-class classifiers for media data classification, Pattern Recognit. 60 (2016) 585–595.
[32] D.M. Tax, P. Laskov, Online SVM learning: from classification to data description and back, in: Neural Networks for Signal Processing, 2003. NNSP'03. 2003 IEEE 13th Workshop on, IEEE, 2003, pp. 499–508.
[33] J. Hamidzadeh, N. Namaei, Belief-based chaotic algorithm for support vector data description, Soft Comput. (2018) 1–26.
[34] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository, 2017. http://archive.ics.uci.edu/ml.
[35] Q.D. Tran, P. Liatsis, User-specific fusion using one-class classification for multimodal biometric systems: Boundary methods, in: 2013 Sixth International Conference on Developments in eSystems Engineering, IEEE, 2013, pp. 276–280.
[36] F. Sohrab, J. Raitoharju, M. Gabbouj, A. Iosifidis, Subspace support vector data description, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 722–727.
[37] N. Kwak, Nonlinear projection trick in kernel methods: an alternative to the kernel trick, IEEE Trans. Neural Netw. Learn. Syst. 24 (12) (2013) 2113–2119.
[38] A. Iosifidis, M. Gabbouj, Nyström-based approximate kernel subspace learning, Pattern Recognit. 57 (2016) 190–197.
[39] T. Elgamal, M. Hefeeda, Analysis of PCA algorithms in distributed environments, arXiv:1503.05214 (2015).
[40] S. Zheng, Smoothly approximated support vector domain description, Pattern Recognit. 49 (2016) 55–64.
[41] A. Sharma, K.K. Paliwal, S. Imoto, S. Miyano, Principal component analysis using QR decomposition, Int. J. Mach. Learn. Cybern. 4 (2013).
[42] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via bipartite graph, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2750–2756.
[43] L.S. Lopes, L.M. Camarinha-Matos, Feature transformation strategies for a robot learning problem, in: Feature Extraction, Construction and Selection, Springer, 1998, pp. 375–391.
[44] L.A. Kurgan, K.J. Cios, R. Tadeusiewicz, M. Ogiela, L.S. Goodenday, Knowledge discovery approach to automated cardiac SPECT diagnosis, Artif. Intell. Med. 23 (2) (2001) 149–169.
[45] F. Sohrab, J. Raitoharju, M. Gabbouj, A. Iosifidis, ms-svdd (github repository), 2020. https://github.com/fahadsohrab/mssvdd.git.

Fahad Sohrab is a PhD student in Unit of Computing Sciences, Tampere University, Finland. He received his MS degree in Electronics Engineering from Sabanci University, Istanbul, Turkey in 2016. His research interests include machine learning, pattern recognition, and anomaly detection.

Jenni Raitoharju is a Senior Research Scientist in Programme for Environmental Information at Finnish Environment Institute. Finland. She received her PhD in Information Technology in Tampere University of Technology, Finland in 2017. Her current projects deal with machine learning and pattern recognition in applications such as biomonitoring and autonomous systems.

Alexandros Iosifidis received his PhD degree in Informatics from the Aristotle University of Thessaloniki in 2014. He is an Associate Professor of Machine Learning at Aarhus University, Denmark. His research interests include statistical machine learning and artificial neural networks with applications in Computer Vision and time-series analysis problems.

Moncef Gabbouj received his MS and PhD degrees in electrical engineering from Purdue University, in 1986 and 1989, respectively. Dr. Gabbouj is Professor of Signal Processing at the Department of Computing Sciences, Tampere University, Finland. His research interests include Big Data analytics, multimedia analysis, artificial intelligence, machine learning, and pattern recognition.