Pattern Anal Applic
DOI 10.1007/s10044-017-0646-3

THEORETICAL ADVANCES

Adaptive three-phase support vector data description

M. Rahmanimanesh[1] · J. A. Nasiri[2] · S. Jalili[3] · N. Moghaddam Charkari[3]

Received: 29 January 2016 / Accepted: 3 August 2017
© Springer-Verlag London Ltd. 2017

Abstract We add a new phase, called reforming phase, to support vector data description (SVDD) between the training and testing phases. The reforming phase enables us to reconsider the SVDD's assumption of the uniformity of features in calculating the distance of an object to the center of hypersphere. In the reforming phase, the features are assumed as a group of experts who have different impacts in overall outlier detection. In doing so, the proportion of each feature in the distance of an object to the center of hypersphere is specified. Subsequently, the opinions of the experts about the label of the corresponding object are determined based on these measured proportions. By using different group decision-making methods for aggregating the opinions of the experts, a large variety of new models are obtained based on one SVDD's trained model. Specially, we utilize a kind of ordered weighted averaging operator as group decision-making method and introduce *c*DFS-SVDD based on this method. *c*DFS-SVDD performs runtime feature selection and calculates the distance of an object to the center of hypersphere dynamically at test time based on these selected features. We apply the method to the anomaly detection problem in mobile ad hoc networks as well as two UCI datasets by which the performance of SVDD improves significantly in separating the target and outlier objects.

Keywords Support vector data description (SVDD) · Group decision making · Ordered weighted averaging (OWA) · Mobile ad hoc networks (MANETs)

1 Introduction

One-class classification aims to make a description of a target set of objects and to detect which new objects resemble this training set [1–4]. The difference with conventional classification is that in the training process of one-class classification problems, only objects of one class are available. The objects from this class are called the target objects, and all other objects are called the outlier objects. The one-class classification also referred as data description, outlier detection, novelty detection, and anomaly detection according to the applications to which one-class classification has been applied. In one-class classification problems, the classification process must rely on one-class classifiers that can train specifications of one existing class. Tax [1] grouped one-class classifiers into three types, namely the density methods (e.g., mixture of Gaussian models [5] and Parzen density estimator [6]), the boundary methods (e.g., nearest neighbor [7], one-class support vector machine [8], and support vector data description [9, 10]), and the reconstruction methods (e.g., k-means [11], learning vector quantization [12], principal component analysis [13], and self-organizing maps [14]).

✉ M. Rahmanimanesh
rahmanimanesh@semnan.ac.ir

J. A. Nasiri
j.nasiri@irandoc.ac.ir

S. Jalili
sjalili@modares.ac.ir

N. Moghaddam Charkari
moghadam@modares.ac.ir

[1] Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Islamic Republic of Iran

[2] Iranian Research Institute for Information Science and Technology (IRANDOC), Tehran, Islamic Republic of Iran

[3] Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Islamic Republic of Iran

Published online: 20 August 2017

https://freepaper.me/t/۳۸۸۴۳۸ : خودت ترجمه کن

In all one-class classification methods two distinct elements should be specified. The first one is a measure for the distance $d(x)$ or resemblance $p(x)$ of an object x to one existing class. The second one is a threshold θ on this distance or resemblance. For prediction, a new object x is labeled as normal when the distance to the target class is smaller than the threshold $[d(x) < \theta]$ or when the resemblance is larger than the threshold $[p(x) > \theta]$. The one-class classifiers can be utilized individually or by any combination methods to better describe the specification of the target class [15–21].

The support vector data description (SVDD) proposed by Tax and Duin [9, 10] tries to construct a boundary with minimal volume around the target data. In the simplest case, a hypersphere is constructed which contains all target objects, but to minimize the chance of accepting outliers, the volume of this hypersphere is minimized. Inspired from SVM [8], the decision boundary of SVDD is described by a number of target objects called support vectors. In general, the hypersphere model is not flexible enough to give a good description of the target class [22]. Analogous to SVM, SVDD offers the ability to transform the data to a new, high-dimensional feature space using kernel functions by which the more flexible descriptions are obtained.

Several extensions, pre-processing, and post-processing methods on SVDD have been proposed in the literature [22–24] to obtain more flexible descriptions and to improve the performance of SVDD. The aim of this paper is also to change the trained boundary of SVDD in order to better separate the target and outlier objects. In contrast to traditional SVDD which assumes that the features have similar impacts in measuring the distance of an object to the center (and in other words, in outlier detection), we consider different impacts of the features in distance measuring (outlier detection). For this purpose, we add a new phase, called reforming phase, between the training phase and the testing phase of SVDD such that it can construct new decision boundaries based on one SVDD's trained support vectors. In doing so, a technique called distance decomposition is introduced. The idea behind feature-based distance decomposition of SVDD is to specify the proportion of each feature in the distance of an object to the center of hypersphere. After that, each feature is considered as an expert who participates in outlier detection, where its opinion about the label of the corresponding object is shaped based on its measured proportion. By using different group decision-making methods [25–27] for aggregating the opinions of the experts, a large number of (maybe infinite) new decision boundaries are obtained.

The main contribution of the paper is as follows:

- We introduced the concept of distance decomposition of SVDD, mapping SVDD to group decision-making problem, and considering different roles of the features in outlier detection task. Note that the superiority of one feature to another feature is determined at runtime for each test object case by case.
- Unlike all other related works that change the SVDD's algorithm and its decision boundary based on training dataset, our proposed method is the first effort that changes the decision boundary of SVDD after training time and based on test objects. Indeed, we can obtain different decision boundary for each test object case by case.

The remainder of the paper is organized as follows. In Sect. 2, we survey some extensions, pre-processing, and post-processing methods proposed on SVDD. In Sect. 3, we present our method for mapping support vector data description to group decision-making problem in the reforming phase. In Sect. 4, we introduce a case study by which we examine our method and also show the results of experiments. In Sect. 5, we utilize the proposed method to examine UCI datasets. Finally Sect. 6 closes the paper with our conclusions.

2 SVDD and its extensions

Support vector data description (SVDD) [9, 10] is a SVM-based one-class classifier that constructs a minimum hypersphere around the target data in a d-dimensional feature space which is characterized by a center $a = (a_1, a_2, \ldots, a_d)$ and a radius R. The distance from objects to the center a should not be strictly smaller than R^2, but larger distances should be penalized. Inspired from SVM, the boundary of hypersphere is determined by a number of target objects called support vectors.

Let $D = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ be the target dataset that contains n objects of dimension d. The problem of finding the minimum hypersphere around the target data can be formulated as:

$$\begin{aligned} &\min R^2 + C \sum_i \xi_i \\ &\text{subject to} \|\mathbf{x}_i - a\|^2 \leq R^2 + \xi_i, \xi_i \geq 0, \forall i, \end{aligned} \tag{1}$$

where $\mathbf{x}_i$s are objects of the training dataset, the slack variable ξ_i is a penalty for object $\mathbf{x}_i$ laid outside the decision boundary, and the regularization parameter C gives the trade-off between the volume of the description and the errors. By introducing Lagrange multipliers, this problem can be transformed into minimizing the following function:

$$L = \sum_i \alpha_i(\mathbf{x}_i.\mathbf{x}_i) - \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i.\mathbf{x}_j), \tag{2}$$

where α_is are Lagrange multipliers with the constraints $0 \leq \alpha_i \leq C, \sum_i \alpha_i = 1, \sum_i \alpha_i x_i = a$.

The minimization of L with the mentioned constraints is a quadratic programming problem that finds the optimal values for the Lagrange multipliers α_i. When an object $\mathbf{x}_i$ satisfies the inequality $\|\mathbf{x}_i - a\|^2 < R^2 + \xi_i$, the corresponding Lagrange multiplier will be zero ($\alpha_i = 0$). For objects satisfying the equality $\|\mathbf{x}_i - a\|^2 = R^2 + \xi_i$ the Lagrange multiplier will become unequal zero ($\alpha_i > 0$). Only objects

https://freepaper.me/t/۳۸۸۴۳۸ : ترجمیار خودت ترجمه کن

$\mathbf{x}_i$ with $\alpha_i > 0$ are needed in the description and therefore be called the *support vectors* of the description (SV's) [9, 10].

In SVDD, the distance of an input object z from the center of the hypersphere is calculated as:

$$\left(D_{\text{SVDD}}(z)\right)^2 = \|z-a\|^2 = K(z,z) - 2\times\sum_i \alpha_i K(z,\mathbf{x}_i) + \sum_i\sum_j \alpha_i\alpha_j K(\mathbf{x}_i,\mathbf{x}_j), \tag{3}$$

where K is the kernel function, $\mathbf{x}_i$ is *ith* support vector, and α_i is the Lagrange multiplier of the support vector $\mathbf{x}_i$.

In the case of linear kernel, we have $K(X,Y) = (X.Y)$. So, $(D_{\text{SVDD}}(z))^2$ is re-formulated as:

$$(D_{\text{SVDD}}(z))^2 = \|z-a\|^2 = (z.z) - 2\times\sum_i \alpha_i(z.\mathbf{x}_i) + \sum_i\sum_j \alpha_i\alpha_j(\mathbf{x}_i.\mathbf{x}_j). \tag{4}$$

The radius of the hypersphere is determined as:

$$R^2 = (\mathbf{x}_l.\mathbf{x}_l) - 2\times\sum_i \alpha_i(\mathbf{x}_l.\mathbf{x}_i) + \sum_i\sum_j \alpha_i\alpha_j(\mathbf{x}_i.\mathbf{x}_j), \tag{5}$$

when the selected support vector $\mathbf{x}_l$ is on the boundary (i.e., $\alpha_l < C$). Accordingly, decision procedure for SVDD is determined as:

$$f_{\text{SVDD}}(z) = I((D_{\text{SVDD}}(z))^2 \le R^2), \tag{6}$$

where the indicator function I is defined as:

$$I(A) = \begin{cases} \text{target} & \text{if A is true,} \\ \text{outlier} & \text{otherwise.} \end{cases} \tag{7}$$

Many research efforts have been done on SVDD to obtain a better description of data. Tax and Juszczak [22] proposed kernel principal component analysis (kernel PCA) as a preprocessing method for SVDD. The data are projected onto the principal components and transformed to a new dataset with zero mean and unit variance in all dimensions. The transformed data can then be better described by the SVDD.

Lee et al. [24] proposed a density-induced SVDD (D-SVDD) to reflect the density distribution of a dataset by introducing the concept of relative density degree for each data point which represents how dense is the region of that data point. By using density-induced distance measurements for target data based on the proposed relative degrees, D-SVDD can shift the center of hypersphere to the denser region based on the assumption that there are more data points in a denser region. They proposed two methods to extract the relative density degree for each data point, the nearest neighborhood method, and the Parzen window method.

Fuzzy support vector machine (FSVM) [28, 29] deals with the situations in which each input object may not be fully assigned to one of two target or outlier classes. FSVM assigns a membership degree to each object and reformulates the SVM in such a way that different objects can have different impacts in the construction of decision boundary. The proposed method causes the SVM to be robust against the noises in data objects. Inspired from fuzzy SVM, Zhang et al. [30] proposed fuzzy SVDD by assigning a weight to each data point which represents fuzzy membership degree to the target data (or degree of importance) computed by the improved possibilistic c-means method. Forghani et al. [31] proposed an extension of fuzzy SVDD in which the features of training objects are fuzzy numbers.

Liu et al. [32] also handle the problem of SVDD-based outlier detection in the presence of uncertain data. Their proposed approach operates in two steps. In the first step, they generate a pseudo-training dataset by assigning a confidence level to each input object. In the second step, they incorporate the generated confidence score for each object into the SVDD's training process. By introducing a confidence score in the training phase, each object contributes differently to the construction of the decision boundary.

Huang et al. [33] considered the differences between the objects in the target dataset and proposed two-class SVDD (TC-SVDD) for the situations in which the target dataset contains two class of objects and each class of objects needs to be described simultaneously. GhasemiGol et al. [34] found a minimal hyperellipse around the target data instead of hypersphere.

Efficient SVDD (E-SVDD) [35] is proposed to improve the prediction speed of SVDD. E-SVDD assumes that the target data contain more than one cluster, and then assigns a unique pre-image to each cluster. If a target training dataset contains c clusters, the E-SVDD first finds the centroid points for these clusters in the feature space and then determines the pre-images for these centroid points. Finally, it uses the expansion of the images of these pre-images to approximate the center of SVDD. Since the most real datasets only contain at most a few clusters, the E-SVDD only needs a few points to represent the center of SVDD and then improves the test speed of SVDD. Compared to fast SVDD [36], E-SVDD obtains the slower prediction speed but better prediction performance.

Guo et al. [23] proposed a post-processing method that constructs a new decision boundary based on the SVDD boundary by derivation of the distance between an object in the input space and its nearest boundary point. The proposed method builds new boundary a distance away from the boundary points such that the boundary still follows the shape of the target distribution but allows more target objects to be accepted.

https://freepaper.me/t/۳۸۸۴۳۸ : خودت ترجمه کن ترجمیار

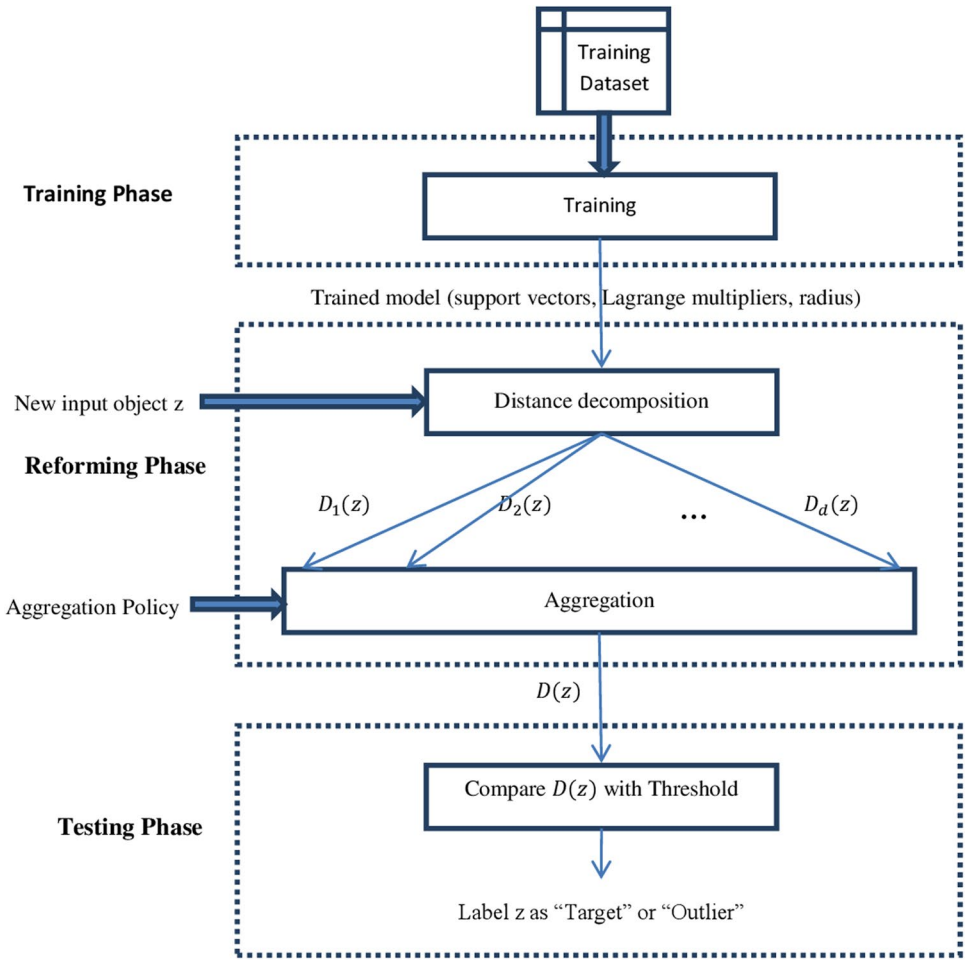

Fig. 1 Three-phase support vector data description

Note that all the related works, as a pre-processing or a post-processing method, change the SVDD's algorithm and its decision boundary based on the training dataset. The approach presented in this paper is different from those of the previous ones. To our best knowledge, this is the first effort that changes the decision boundary of SVDD after training time, regardless of training dataset, and based on any assumption on the distribution of test data. This is done by transforming support vector data description to group decision-making problem by which a large variety of new SVDD models can be obtained.

3 Three-phase SVDD

The aim of our proposed approach is not only to obtain a tighter boundary around the training data, but is also to reform the decision boundary of SVDD after training time in order to improve the performance of SVDD in accepting the target objects and rejecting the outlier objects.

The block diagram of the proposed method is shown in Fig. 1 in which we add a new phase, called reforming phase, between the training and testing phases of SVDD. In the reforming phase, we use the trained model of SVDD to construct new decision boundaries by reconsidering the SVDD's assumption of the uniformity of features in classification task. In this phase, we reform the trained boundary of SVDD according to the aggregation policy. In doing so, we first determine the proportion of each feature in the distance of an object to the center of hypersphere [in Fig. 1, $D_i(z)$ is the proportion of feature i for an input object z, and d is the number of features]. Subsequently, each feature is assumed to be an expert who participates in outlier detection, where its opinion about the label of the corresponding object is shaped based on its measured proportion. Several methods have been proposed in the literature for group decision making [25–27], which can be applied in the aggregation. By utilizing these different methods (i.e., by assigning different roles to the features in outlier detection), the reformation will lead to new boundary shapes.

3.1 Feature-based distance decomposition of SVDD

The idea behind distance decomposition of SVDD is to specify the proportion of each feature in the distance of an

https://freepaper.me/t/۳۸۸۴۳۸ خودت ترجمه کن : ترجمیار

object to the center of hypersphere. Let $z = (z_1, z_2, \ldots, z_d)$ be a d-dimensional object in the input space. Distance decomposition for kth feature, $k = 1, \ldots, d$, of the object z using SVDD is carried out as:

$$(D_{\text{SVDD},k}(z))^2 = {z_k}^2 - 2 \times z_k \times \sum_i (\alpha_i \times \mathbf{x}_{ik}) + \sum_i \sum_j (\alpha_i \times \alpha_j \times \mathbf{x}_{ik} \times \mathbf{x}_{jk}), \tag{8}$$

where $\mathbf{x}_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{id})$ is ith support vector, and α_i is the Lagrange multiplier of the support vector $\mathbf{x}_i$.

Now, feature k is an expert whose opinion about the object z is shaped based on $(D_{\text{SVDD},k}(z))^2$. Note that $(D_{\text{SVDD}}(z))^2 = \sum_{i=1}^{d} (D_{\text{SVDD},i}(z))^2$.

3.2 WA-SVDD and OWA-SVDD

We examine two special methods, namely weighted average (WA) and ordered weighted average (OWA) [25], proposed for group decision making and define WA-SVDD and OWA-SVDD based on these methods. In WA-SVDD and OWA-SVDD, instead of calculating sum of the proportions of features (which is performed in traditional SVDD), we calculate weighted sum and ordered weighted sum of the proportions, respectively.

Let $D_{\text{SVDD},i}(z)$ be the proportion of feature i in the distance of object z to the center of hypersphere. The WA-SVDD assigns a fixed weight to each feature. So, we have:

$$(D_{\text{WA-SVDD}}(z))^2 = \sum_{i=1}^{d} w_i \times (D_{\text{SVDD},i}(z))^2, \tag{9}$$

where w_i is the weight of ith feature, and $\sum_{i=1}^{d} w_i = d$.

OWA-SVDD uses OWA operator to signify those features that make more (or less) deviations from the center of hypersphere. So, we have:

$$(D_{\text{OWA-SVDD}}(z))^2 = \sum_{i=1}^{d} w_i B_i, \tag{10}$$

where B_i is the ith largest element of the bag $<(D_{\text{SVDD},1}(z))^2, \ldots, (D_{\text{SVDD},d}(z))^2>$, w_i is the weight of B_i, and $\sum_{i=1}^{d} w_i = d$.

By using different weighting vectors, different boundary shapes for WA-SVDD and OWA-SVDD are obtained. Note that in both WA-SVDD and OWA-SVDD, if we use $W = (1, 1, \ldots, 1)$ weighting vector, the classifiers become idempotent and we have:

$$(D_{\text{SVDD}}(z))^2 = (D_{\text{WA-SVDD}}(z))^2 = (D_{\text{OWA-SVDD}}(z))^2. \tag{11}$$

We can use different decision procedures for WA-SVDD and OWA-SVDD. For example, decision procedure for WA-SVDD (and OWA-SVDD) can be defined as:

$$f_{\text{WA-SVDD}}(z) = I((D_{\text{WA-SVDD}}(z))^2 \leq \theta^2), \tag{12}$$

where θ^2 is calculated as:

$$\theta^2 = \frac{1}{n} \sum_{i=1}^{n} (D_{\text{WA-SVDD}}(\mathbf{x}_i))^2, \tag{13}$$

where n is the number of support vectors, and $\mathbf{x}_i$ is ith support vector. Alternatively, the threshold can be tuned such that the new classifier has the same false-positive rate as the SVDD model, meaning that it rejects the same number of training objects by the constructed boundary.

3.3 An illustrative example

Figure 2 shows the reformed boundaries of SVDD according to three different types of outlier objects. In the figure, the two-dimensional training dataset D is shown in pink color and the outlier objects are shown in blue color. Figure 2a shows three different models obtained by using SVDD on the dataset D with three different ν parameters. The boundaries of three models are shown in brown, red, and green colors for ν equal to 0.26, 0.4, and 0.66, respectively. In the example, the closer the false-positive rate (FP) to 0 and the detection rate (DR) to 1 for a classifier, that classifier is more powerful in accepting the target objects and rejecting the outlier objects.

(a) The training dataset and three different SVDD models
The pink points: training dataset
The blue points: outlier objects

The boundary of SVDD with ν=0.26, R= 0.858836
The boundary of SVDD with ν=0.4, R= 0.769415
The boundary of SVDD with ν=0.66, R= 0.582477
(b) The boundary of WA-SVDD with weighting vector $w = (1.55, 0.45)$, and θ=0.71
(c) The boundary of OWA-SVDD with weighting vector $w = (1.9, 0.1)$, and θ= 1.0564
(d) The boundary of OWA-SVDD with weighting vector $w = (0.1, 1.9)$, and θ= 0.4437

(b)						(c)						(d)					
FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR
0.4080	0.932	0.6578	0.999	0.4069	1	0.4080	0.997	0.2611	0.9354	0.2611	0.9941	0.4080	0.903	0.6578	0.979	0.3957	0.9967

https://freepaper.me/t/۳۸۸۴۳۸ : خودت ترجمه کن

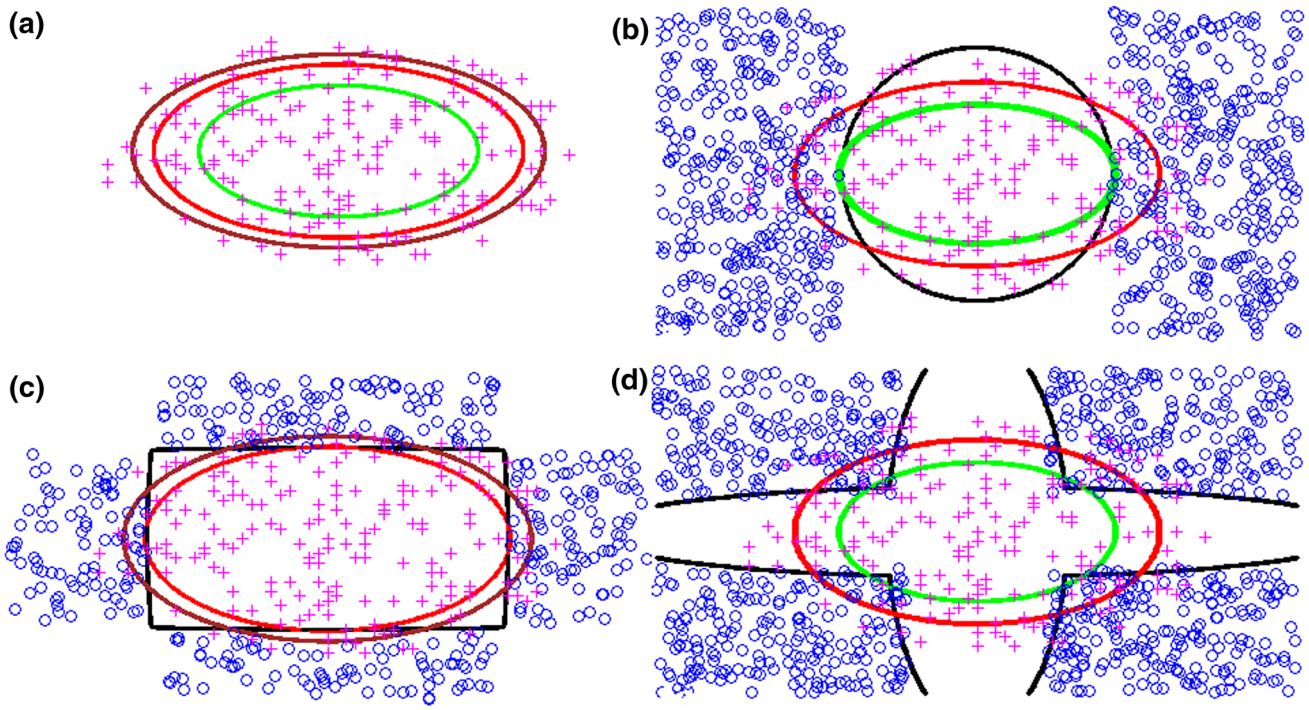

Fig. 2 The models obtained by using SVDD, WA-SVDD, and OWA-SVDD (color figure online)

In Fig. 2b, we aim to classify a special type of outlier objects that are shown in blue color by three classifiers, the SVDD with red boundary, the SVDD with green boundary, and the WA-SVDD with the weighting vector $w = (1.55, 0.45)$, i.e., the black boundary. In this case, if we use SVDD with red boundary, we obtain the detection rate equal to 0.9322 at the expense of 0.4080 false-positive rate. Subsequently, if we want to increase the detection rate to 0.9990 by SVDD, we must utilize SVDD with green boundary at the expense of 0.6578 false-positive rate, while, by WA-SVDD, the classifier obtains the detection rate approximately equal to the SVDD with green boundary (1) and the false-positive rate approximately equal to the SVDD with red boundary (0.4069).

Furthermore, in Fig. 2c, the OWA-SVDD classifier with the weighting vector $w = (1.9, 0.1)$, i.e., the black boundary, has detection rate approximately equal to the SVDD with red boundary and the false-positive rate approximately equal to the SVDD with brown boundary. Also, in Fig. 2d, the OWA-SVDD classifier with the weighting vector $w = (0.1, 1.9)$, i.e., the black boundary, has detection rate approximately equal to the SVDD with green boundary and the false-positive rate approximately equal to the SVDD with red boundary.

As it is shown in Fig. 2, for all three types of outlier objects, the decision boundary of one trained classifier is reformed according to the distribution of the test objects. This reformation enables us to change the target objects that are rejected by the classifier in order to better reject a specific type of outliers, and leads to the better performance by improving the detection rates for the same false-positive rates or by decreasing the false-positive rates for the same detection rates.

Specially, as it is shown in Fig. 2b, the WA-SVDD constructs a *hyperellipse* around the training data and allows the target objects to have more distance in the direction of features that have fewer weights. Furthermore, in OWA-SVDD, according to Fig. 2c, if we use descendant weights, the target objects are allowed to be farther from the center when the features of the objects deviate uniformly from the center (this can be interestingly stated as: *When the experts have different opinions, they should pay the penalties*, or alternatively *when the experts have relatively the same opinions, they will gain the awards*). In this case, a square-shaped model can be obtained by the classifier. Moreover, according to Fig. 2d, if we use ascendant weighting vector, the target objects are allowed to have more distance in the direction of Cartesian axes (this can be stated as: *The opinions of those experts are more important who are more optimistic*). In this case, a fan-shaped model can be obtained by the classifier. Note that these exciting models are only obtained by WA-SVDD and OWA-SVDD, whereas, if we use other group decision-making methods, the more and more different models will be obtained.

https://freepaper.me/t/۳۸۸۴۳۸ خودت ترجمه کن : ترجمیار

4 Experiments

To evaluate the proposed method, we apply the method to the anomaly detection problem in mobile ad hoc networks with AODV routing protocol. AODV assumes that all nodes of the network can be trusted to perform their tasks truthfully, and as such, is vulnerable to several attacks and misbehavior. Anomaly detection refers to detecting the behavior of nodes that do not conform to a pre-defined normal behavior (the trained model), where the deviations from the network's normal behavior are considered as anomalies or attacks[1] (i.e., outliers).

In our proposed method, each node of the network uses a time slot to analyze the state of the network. In each time slot, the network behavior is expressed by a 40-dimensional feature vector in which each feature is measured to describe a part of AODV protocol characteristics. We assume that the mobile ad hoc network is flat and completely distributed. Each node collects its own data in the network, and there is no need for monitoring the behavior of a node by its neighbors.

We begin by training the SVDD on the training dataset which contains only normal objects. In the method, the training is offline and the trained model is stored in each node of the network for anomaly detection. Subsequently, for online network traffic (i.e., the test data), we measure deviations from the trained model by the transformed SVDD to identify the attack and normal objects according to the measured deviations.

4.1 Overview of AODV protocol

AODV [37, 38] is a reactive routing protocol for MANETs, where each node maintains a routing table. AODV uses two mechanisms, namely route discovery and route maintenance. The process of route discovery begins when the source node wants to send a data packet to the destination and there is no valid route for that destination in its routing table. In this case, the source node broadcasts the route request (RREQ) packet in the network. When the RREQ packet reaches a node that knows a route to the destination, the node will unicast the route reply (RREP) packet to the node that it has received the RREQ packet from it, and this action will repeat until the RREP packet reaches the source node. As soon as the source node receives the RREP packet, it can start transmitting data packets. The process of route maintenance begins when a broken link is detected or the next node is inaccessible. In this case the node that detects this will send a route error (RERR) packet to all its active neighbors for that destination.

[1] In this section, we use the terms normal and attack instead of target and outlier, respectively, because of the special application to which we involved.

4.2 Definition of features

Each node uses a time slot to analyze the state of the network according to its own ingoing and outgoing traffic, routing table, packet queue, etc. In each time slot, the network state is expressed by a 40-dimensional object $z = (z_1, z_2, \ldots, z_{40})$, where each feature $z_i, i = 1, \ldots, 40$ is measured to describe a part of AODV protocol characteristics. The defined features are grouped into three categories, traffic-related features, extracted fields of observed packets, and routing table-related features as shown in Table 1.

4.3 Attack implementation

We launch the following attacks on the network to evaluate our proposed method:

1. Blackhole: In this attack, a malicious node tries to pass the networks traffic through itself (by sending fake RREP packets) and then drop any control or data packets that reach it.
2. Denial of service (with the name of DoS): In this attack, a malicious node intentionally uses the resources (bandwidth and energy) of other nodes in the network or makes the target node unavailable. This is done by continuous injection of data packets to the network.
3. RERR fabrication: In this attack, a malicious node can fake some RERR packets, which can lead to the destruction of the main route, and the imposition of overhead to the network.

4.4 Dynamic feature selection by OWA-SVDD

Let $z = (z_1, z_2, \ldots, z_d)$ be a d-dimensional object in the input space. A c-dynamic feature selection ordered weighted average (cDFSOWA) operator of dimension d is a mapping $c\text{DFSOWA}: R^d \to R$ that has an associated d-dimensional weighting vector $W = (w_1, w_2, \ldots, w_d)$, such that:

$$w_i = \begin{cases} \frac{d}{c} & 1 \le i \le c, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

and

$$c\text{DFSOWA}(z) = \sum_{i=1}^{d} w_i B_i, \tag{15}$$

https://freepaper.me/t/۳۸۸۴۳۸ : خودت ترجمه کن ترجمیار

Table 1 Defined features

Feature number	Description
A. Traffic-related features (21 features)	
1–2	Number of sent [packets/control packets]
3–6	Number of sent [RREQ/RREP/RERR/data] packets
7–8	Number of [packets/control packets] received by node
9–12	Number of received [RREQ/RREP/RERR/data] packets
13–16	Number of dropped [RREQ/RREP/RERR/data] packets
17–19	Number of replicated [RREQ/RREP/data] packets received by node
20–21	Number of added packets to queue—max queue length
B. Extracted fields of observed packets (8 features)	
22–23	Total differences between the magnitude of hop count of received [RREQ/RREP] packets and the magnitude of hop count of related entries in the routing table
24–27	[Max/total] differences between the magnitude of sequence number of received [RREQ/RREP] packets and the magnitude of sequence number of related entries in the routing table
C. Routing table-related features (11 features)	
28–29	[Average/max] changes in the magnitude of sequence number of routing table entries
30–31	Number of times that sequence number field of routing table entries updated (for valid routes)—total changes of this field
32–33	Number of times that sequence number field of routing table entries fixed (for invalid routes)—total changes of this field
34–35	Number of times that hop count field of routing table entries [fixed (for invalid routes)/updated (for valid routes)]
36–39	Number of successful routing table lookups for [valid/invalid] routes—number of unsuccessful routing table lookups, rate of routing table successful lookups
40	Number of valid routes div by number of invalid routes

where $c, 1 \leq c \leq d$, is a cutting parameter specified by user, and B_i is the *ith* largest element of the bag $<z_1, \ldots, z_d>$. Note that $\sum_{i=1}^{d} w_i = d$, and if $c = d$, then $W = (1, 1, \ldots, 1)$ and cDFSOWA becomes idempotent.

We define cDFS-SVDD classifier such that it uses cDFSOWA operator for group decision making. cDFS-SVDD is a runtime feature selection operator. The number of c features that make more deviations from the center is selected dynamically for each test object. In other words, cDFS-SVDD calculates the distance of an object to the center of hypersphere only based on those dynamically selected features, and other features will be disregarded.

We define cDFS-SVDD based on the experiences that we gained from the anomaly detection task in mobile ad hoc network (MANET):

1. The distribution of attacks (the outlier data) on AODV routing protocol in MANET does not conform to the distribution of the normal behavior (the target data).
2. Although the SVDD considers all "non-target" objects as "outliers" and aims to reject all outlier objects, the outlier objects have different types and patterns. Each type of outliers (attacks) has its own pattern and makes more deviation in the magnitude of some specific features, but these features are different for each type of outliers. In other words, each attack has its own "important features" which are different from the other attacks.
3. We cannot specify these important features anyway in the training time. Amazingly, the features that seem more useless in the training time may be the more important features in the testing time for detection of attacks.

Based on these observations, we found out that the application strongly needs a classifier that makes decision based on a dynamic subset of features. So, we have altered the SVDD and added the reforming phase to SVDD in which we map the support vector data description to group decision-making problem. By using ordered weighted average in the reforming phase as group decision-making method, we define cDFS-SVDD that can determine dynamically the more important and valuable features in the testing time for each type of attacks and for each input object case by case, and only use these features in data description and outlier detection task.

4.5 Data scaling

The performance of SVDD critically depends on the scaling of data. In this paper for all training and testing objects, we

https://freepaper.me/t/۳۸۸۴۳۸ : ترجمیار خودت ترجمه کن

use a well-known min–max data scaling method performed by Library for Support Vector Machines (LIBSVM) [39] that transforms data to a new dataset with unit range in all feature directions.

Let $D = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ be the target dataset that contains n objects of dimension d, where $\mathbf{x}_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{id})$, $\check{\mathbf{x}} = (\check{\mathbf{x}}_1, \check{\mathbf{x}}_2, \ldots, \check{\mathbf{x}}_d)$ be the minimum of D, $\hat{\mathbf{x}}$ be the maximum of D, and $\Delta = \hat{\mathbf{x}} - \check{\mathbf{x}} = (\delta_1, \delta_2, \ldots, \delta_d)$. To scale D, we transform D to $D' = [\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_n]^T$, where $\mathbf{x}'_i = (\mathbf{x}'_{i1}, \mathbf{x}'_{i2}, \ldots, \mathbf{x}'_{id})$ as:

$$\mathbf{x}'_{ij} = \frac{\mathbf{x}_{ij} - \check{\mathbf{x}}_j}{\delta_j}, \quad i = 1, \ldots, n, j = 1, \ldots, d. \tag{16}$$

After that, each new test dataset $T = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m]^T$, where $\mathbf{y}_i = (\mathbf{y}_{i1}, \mathbf{y}_{i2}, \ldots, \mathbf{y}_{id})$, is scaled by transformation to $T' = [\mathbf{y}'_1, \mathbf{y}'_2, \ldots, \mathbf{y}'_m]^T$, where $\mathbf{y}'_i = (\mathbf{y}'_{i1}, \mathbf{y}'_{i2}, \ldots, \mathbf{y}'_{id})$ as:

$$\mathbf{y}'_{ij} = \frac{\mathbf{y}_{ij} - \check{\mathbf{x}}_j}{\delta_j}, \quad i = 1, \ldots, m, j = 1, \ldots, d. \tag{17}$$

4.6 Feature sensitivity analysis

Feature sensitivity analysis (FSA) is an offline process that compares two datasets containing the objects of the same feature space. FSA aims to characterize those features of the objects of a test dataset T that make more deviations from the corresponding features of the objects of the training dataset D.

Suppose the training dataset D and the test dataset T are scaled to D' and T', respectively, by the method presented in Sect. 4.5.

Let $\overline{\mathbf{x}'} = (\overline{\mathbf{x}'_1}, \overline{\mathbf{x}'_2}, \ldots, \overline{\mathbf{x}'_d})$ be the mean of D', and $\overline{\mathbf{y}'} = (\overline{\mathbf{y}'_1}, \overline{\mathbf{y}'_2}, \ldots, \overline{\mathbf{y}'_d})$ be the mean of T'. FSA is performed by the calculation of *average feature deviation* for dataset T and feature k, AFD(T, k) as:

$$\mathrm{AFD}(T, k) = \overline{\mathbf{y}'_k} - \overline{\mathbf{x}'_k}, \quad k = 1, \ldots, d. \tag{18}$$

4.7 Simulation environments

We use Network Simulator NS-2 version 2.34 [40] to run MANET simulations. Our experiments are based on 50 wireless mobile nodes distributed in a 1000 × 1000 m area, which follow the random way-point mobility model with the maximum speed of 5 m/s and a pause time of 10 s. Network traffic type is constant bit rate (CBR), data packet size is 512 bytes, routing protocol is AODV, and the maximum number of connections is 40 packets per second. Simulation time is 3000 s, and each time slot is 30 s. Regular nodes normally perform routing as well as anomaly detection. In contrast, each malicious node launches attacks (i.e., blackhole, DoS, and RERR fabrication). The simulations are done on 100 different traffic patterns and movement scenarios, and the average results are shown.

4.8 Simulation results

We now compare the performances of the traditional SVDD and cDFS-SVDD on the application of anomaly detection in MANETs. The ROC curves for three mentioned attacks by SVDD and 4DFS-SVDD (cDFS-SVDD with parameter c equal to 4) are shown in Fig. 3. By 4DFS-SVDD, those 4 features that make more deviations from the saved normal profile are selected dynamically in runtime. These features may be different for each type of attacks (and also for each object case by case). Note that all data have been scaled before training and testing phases by the method presented in Sect. 4.5. Also, for calculating detection rates and false-positive rates, a time slot is labeled as "attack" if five (or more) nodes of the network label that time slot as "attack."

According to Fig. 3, in summary, the blackhole attack is detected with the highest detection rate and the RERR fabrication attack is detected with the lowest detection rate. Figure 3a shows the ROC curves for blackhole attack by two classifiers using 40 basic features (presented in Table 1), where the attack is detected with the rate near to %100 by the two classifiers. In Fig. 3b, c, the ROC curves for DoS and RERR fabrication attacks are shown, where it shows that for all false-positive rates, the detection rates of DoS attack improve about 0.06 and the detection rates of RouteError fabrication attack improve about 0.1. The AUC of these two classifiers is shown in Table 2 in which 4DFS-SVDD shows better performance than the SVDD on average.

Figure 4 shows AFD(T, k) for one normal test dataset and three attack datasets (based on 40 basic features). As it is shown, each attack makes more deviations in the magnitude of some specific features, but these features are different in the case of different attacks. Indeed, 4DFS-SVDD selects these important features dynamically and only uses these features in distance measuring (in contrast to the existing feature selection methods that select a fixed subset of features statically).

4.9 Utilizing cDFS-SVDD in principal components space

To show the effectiveness of our proposal, we utilize 4DFS-SVDD in principal components (PCs) space. In doing so, we transform data to PCs space by using Tax pre-processing method [22] and obtain 35-dimensional feature space (by selecting the first 35 features in the transformed space). Table 3 compares AUC of SVDD and 4DFS-SVDD in

https://freepaper.me/t/۳۸۸۴۳۸ : ترجمه خودت ترجمه کن

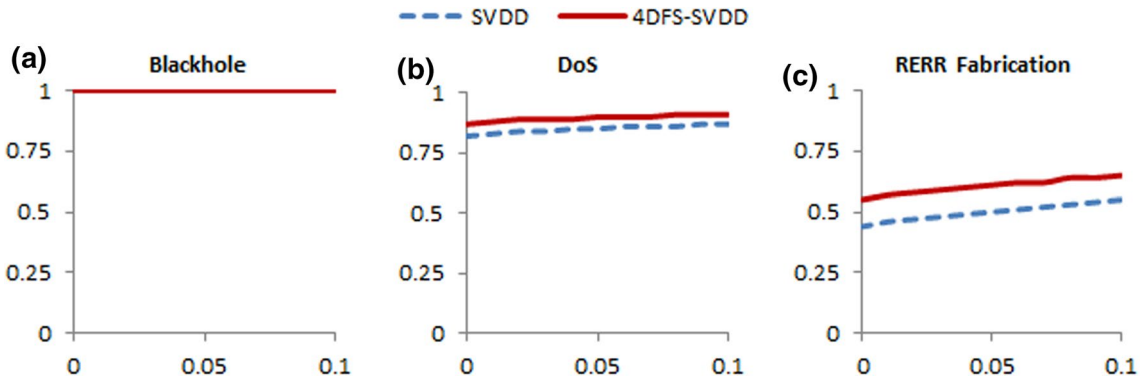

Fig. 3 ROC curves for SVDD and 4DFS-SVDD on 40 basic features *X*-axis: false-positive rate, *Y*-axis: detection rate

Table 2 Comparison of the AUC of SVDD and 4DFS-SVDD on 40 basic features

	Blackhole mean/SD	DoS mean/SD	RERR fabrication mean/SD	Average mean/SD
SVDD	0.1/0	0.0848/0.0026	0.0494/0.0044	0.0781/0.0023
4DFS-SVDD	0.1/0	0.0894/0.0014	0.0602/0.0041	0.0832/0.0018

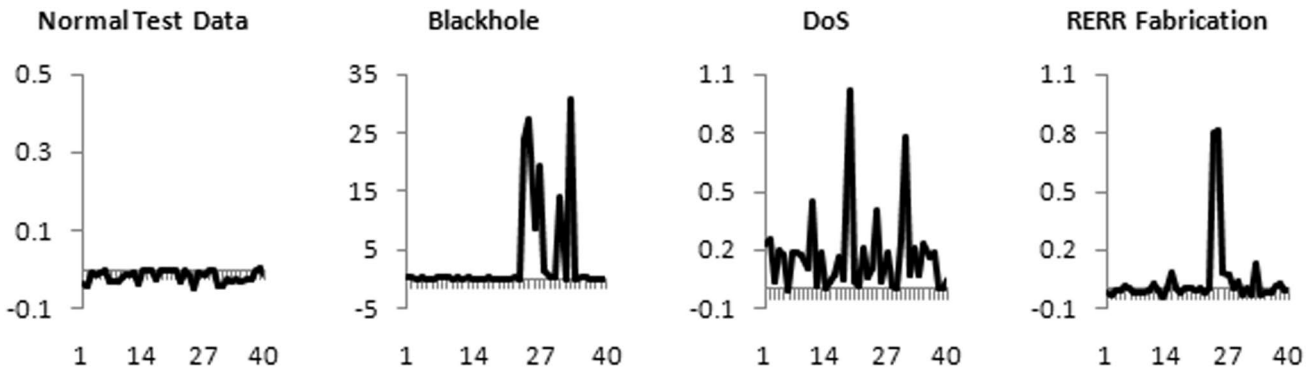

Fig. 4 FSA for one normal test dataset and three attack datasets *X*-axis: feature number, *Y*-axis: AFD(T, k)

Table 3 Comparison of the AUC of SVDD and 4DFS-SVDD on 35 features obtained by using Tax pre-processing method [22]

	Blackhole mean/SD	DoS mean/SD	RERR fabrication mean/SD	Average mean/SD
SVDD	0.1/0	0.0905/0.0008	0.0781/0.0024	0.08953/0.0010
4DFS-SVDD	0.1/0	0.0936/0.0008	0.0803/0.0026	0.0913/0.0011

35-dimensional feature spaces. As it is shown, 4DFS-SVDD also improves AUC of SVDD in PCs space.

In order to clarify the improvements, we perform feature sensitivity analysis on one normal and three attack datasets in PCs space. As it can be seen in Fig. 5, the non-uniformity of the feature deviations from the saved normal profile is also hold in PCs space, and this is the evidence for why 4DFS-SVDD performs better than SVDD even using Tax pre-processing method. According to Fig. 5, it is clear that the first few dimensions are crucial for normal patterns, but this is not necessarily true for each type of attacks.

Table 4 compares the AUC of SVDD on 4-dimensional feature space (the first 4 features in PCs space) and 4DFS-SVDD on 35-dimensional feature space. As it can be seen, 4 dynamically selected features perform significantly better than 4 statically selected features in detecting anomalies.

https://freepaper.me/t/۳۸۸۴۳۸ : خودت ترجمه کن ترجمیار

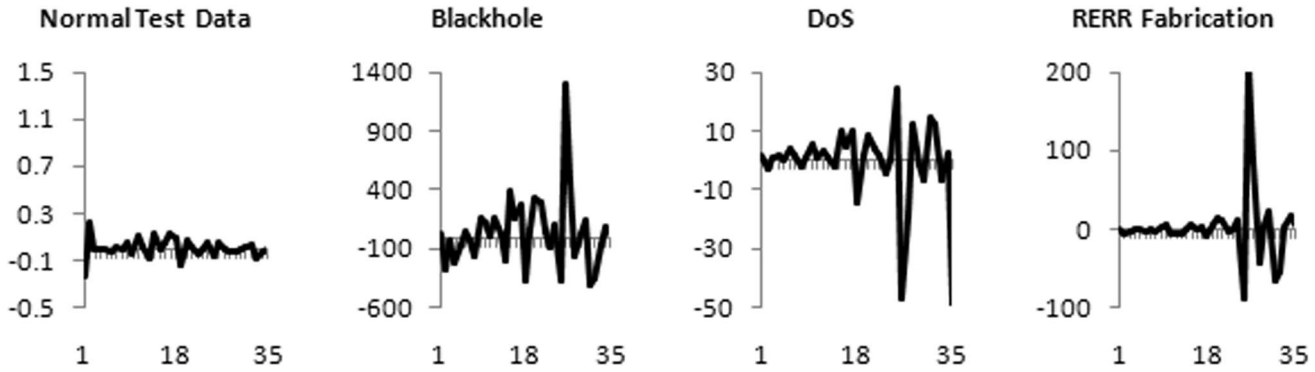

Fig. 5 FSA for 35 features when data are transformed to PCs space by Tax pre-processing method [22] *X*-axis: feature number, *Y*-axis: AFD(T, k)

Table 4 Comparison of the AUC of 4 dynamically selected features (from 35 features) and 4 statically selected features on PCs space

	Blackhole mean/SD	DoS mean/SD	RERR fabrication mean/SD	Average mean/SD
Four dynamically selected features (4DFS-SVDD)	0.1/0	0.0936/0.0008	0.0803/0.0026	0.0913/0.0011
Four statically selected features (SVDD)	0.1/0	0.0531/0.0059	0.052/0.0040	0.06836/0.0033

Table 5 Properties of two standard datasets

No.	Name	No. of samples	No. of features	No. of classes
1	Image segmentation	2100	19	7
2	Wholesale customers	440	8	6

5 Evaluations and discussion

In this section, we apply our proposed method to two standard UCI datasets, i.e., image segmentation and wholesale customer datasets. These datasets are taken from the UCI machine learning repository [41]. Table 5 shows the specifications of these two datasets used for evaluating our proposed method.

As it is shown in Table 5, the image segmentation dataset contains 2100 objects with 19 features in seven different classes. For one-class classification, we use each of Class1, Class3, Class5, and Class7 datasets in turn as target dataset and all other classes as outliers, according to one-against-all strategy. Table 6 shows the performance of *c*DFS-SVDD as compared to SVDD (with RBF kernel) based on true-positive rates (TP) and false-positive rates (FP), where the closer the FP to 0 and the TP to 1 for a classifier, that classifier is more powerful in accepting the target objects and rejecting the outlier objects. Note that in each case, we show the FPs of SVDD/*c*DFS-SVDD for two selected TPs, i.e., TP1 and TP2.

According to Table 6, the performance of *c*DFS-SVDD is always equal to or greater than the performance of SVDD in all four cases of target datasets, meaning that it exhibits less FP for the same TP. The reduction more than 0.05 (%5) of FP for the same TP is shown in bold font in Table 6. For example, if we use Class7 as target object, and other classes as outliers, for TP equal to 0.64, *c*DFS-SVDD can reduce the FP of Class3 about 0.56 as compared to SVDD, i.e., a significant improvement in the performance of classification task.

Figure 6 shows the results of feature sensitivity analysis of five test datasets (Class7 dataset containing test target objects, and Class1, Class2, Class3, Class4 datasets containing four types of test outlier objects). Note that the results of Class5 and Class6 datasets are forbidden because by both SVDD and *c*DFS-SVDD classifiers are classified with the same and low (0) FP. As it is shown in Fig. 6, each type of outliers has its own important features which are different from other types of outliers, the case that *c*DFS-SVDD can be used for classification.

We also used the wholesale customers dataset for comparing the performance of *c*DFS-SVDD and SVDD. The dataset contains 440 objects with eight features in six different classes. For one-class classification task, we use Channel1 data (Region1, Region2, and Region3) in turn as training datasets and Channel2 data (Region1, Region2, and Region3) as test objects. Table 7 compares the performance of the classifiers, which shows a relatively better

https://freepaper.me/t/۳۸۸۴۳۸ : خودت ترجمه کن ترجمیار

Table 6 The comparison of SVDD (with RBF kernel) and *c*DFS-SVDD on image segmentation dataset

Target dataset	TP for target dataset		FP for Class1 SVDD/*c*DFS-SVDD	FP for Class2 SVDD/*c*DFS-SVDD	FP for Class3 SVDD/*c*DFS-SVDD	FP for Class4 SVDD/*c*DFS-SVDD	FP for Class5 SVDD/*c*DFS-SVDD	FP for Class6 SVDD/*c*DFS-SVDD	FP for Class7 SVDD/*c*DFS-SVDD
Class1 (brickface)	TP1	0.82	–	0.05/0.03	0.01/0.01	0/0	0/0	0/0	**0.15/0.1**
	TP2	0.79	–	0.03/0.01	0.01/0.01	0/0	0/0	0/0	0.08/0.04
Class3 (foliage)	TP1	0.87	0.03/0.02	0.33/0.33	–	0/0	0.12/0.09	0/0	0.65/0.65
	TP2	0.84	0.02/0.01	0.31/0.31	–	0/0	**0.11/0.02**	0/0	0.61/0.60
Class5 (path)	TP1	0.87	0/0	0.01/0	0/0	0/0	–	0/0	0/0
	TP2	0.82	0/0	0.01/0	0/0	0/0	–	0/0	0/0
Class7 (window)	TP1	0.64	**0.50/0.04**	0.04/0.04	**0.68/0.12**	**0.32/0**	0/0	0/0	–
	TP2	0.60	**0.44/0.04**	0.04/0.04	**0.58/0.12**	**0.16/0**	0/0	0/0	–

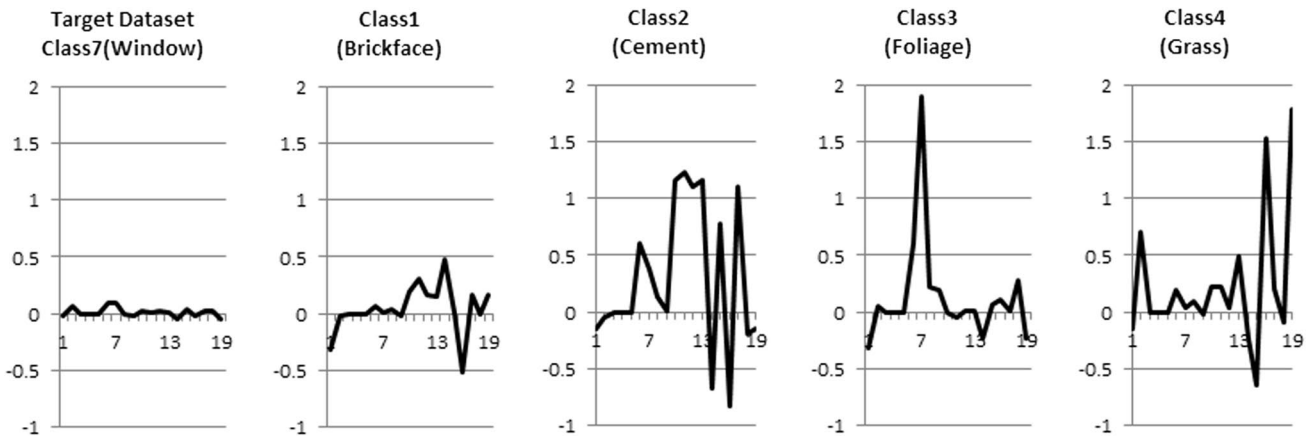

Fig. 6 Feature sensitivity analysis for image segmentation dataset, Class7 as target dataset, and other classes as outliers *X*-axis: feature number, *Y*-axis: AFD(T, k)

performance for *c*DFS-SVDD as compared to SVDD with RBF kernel. The reduction more than 0.05 (%5) of FP for the same TP are shown in bold font in Table 7. For example, if we use Class3 as target object, for TP equal to 0.98, *c*DFS-SVDD can reduce the FP of Class5 about 0.09 as compared to SVDD.

6 Conclusion

In this paper, we mapped the support vector data description to a group decision-making problem through ascribing different roles to features in outlier detection task. In doing so, a technique called feature-based distance decomposition of SVDD is introduced by which the proportion of each feature in making the distance of an object to the center of hypersphere is specified. After that, each feature is considered as an expert who participates in outlier detection task, in which its opinion about the label of corresponding object is shaped based on its measured proportion. By using different group decision-making methods for aggregating the opinions of the experts, a large number of new decision boundaries are obtained in testing time based on one SVDD's trained model. Specially, we examined a specific widely used group decision-making method, the ordered weighted average method, and introduced *c*DFS-SVDD based on this method. *c*DFS-SVDD performs runtime feature selection and calculates the distance of an object to the center of hypersphere dynamically based on c features that have more deviations from the center, and other features will be ignored. The selected features may be different for each object and determined dynamically at runtime. The proposed method can be utilized in the problems for which all features of target objects have the same importance, but the outlier objects have different types and patterns and each outlier object has its own important features which are different

https://freepaper.me/t/۳۸۸۴۳۸ : ترجمه خودت ترجمیار

Table 7 The comparison of SVDD (with RBF kernel) and *c*DFS-SVDD on wholesale customers dataset

Target dataset	TP for target dataset		FP for Class4 (Channel2, Region1) SVDD/*c*DFS-SVDD	FP for Class5 (Channel2, Region2) SVDD/*c*DFS-SVDD	FP for Class6 (Channel2, Region3) SVDD/*c*DFS-SVDD
Class1 (Channel1, Region1)	TP1	0.96	0.33/0.33	0.52/0.52	0.52/0.51
	TP2	0.93	0.33/0.33	**0.52/0.47**	0.51/0.51
Class2 (Channel1 Region2)	TP1	0.96	0.11/0.11	0.05/0.05	0.02/0.02
	TP2	0.92	0.11/0.11	0/0	0.02/0.01
Class3 (Channel1, Region3)	TP1	0.99	0.55/0.55	0.57/0.57	**0.70/0.64**
	TP2	0.98	**0.50/0.44**	**0.57/0.47**	**0.61/0.54**

from the other ones. We applied the method to anomaly detection problem in mobile ad hoc networks as well as two UCI datasets. The results of experiments showed that *c*DFS-SVDD improves the performance of SVDD in detecting the target and outlier objects.

References

1. Tax DMJ (2001) One class classification. Ph.D. thesis, Delft University of Technology
2. Krawczyk B, Wozniak M (2015) One-class classifiers with incremental learning and forgetting for data streams with concept drift. Soft Comput 19(12):3387–3400
3. Krawczyk B, Wozniak M (2015) Incremental weighted one-class classifier for mining stationary data streams. J Comput Sci 9:19–25
4. Nguyen DT, Cios KJ (2015) Rule-based oneclass-ds learning algorithm. Appl Soft Comput 35:267–279
5. Bishop C (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
6. Duda RO, Hart PE, Stork DG (2000) Pattern Classification, 2nd edn. Wiley, New York
7. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185
8. Scholkopf B, Platt JC, Taylor JS, Smola AJ, Williamson RC (2001) Estimating the support of a high dimensional distribution. Neural Comput 13:1443–1471
9. Tax DMJ, Duin RPW (1999) Support vector domain description. Pattern Recognit Lett 20:1191–1199
10. Tax DMJ, Duin RPW (2004) Support vector data description. Mach Learn 54(1):45–66
11. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall, Englewood Cliffs
12. Kohonen T (1995) The handbook of brain theory and neural networks. MIT Press, Cambridge
13. Jolliffe I (2002) Principal component analysis, 2nd edn. Springer, Berlin
14. Kohonen T (2001) Self-organizing maps, 3rd edn. Springer, Berlin
15. Krawczyk B (2015) One-class classifier ensemble pruning and weighting with firefly algorithm. Neurocomput 150:490–500
16. Krawczyk B, Wozniak M, Herrera F (2015) On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. Pattern Recognit 48(12):3969–3982
17. Krawczyk B, Wozniak M, Cyganek B (2014) Clustering-based ensembles for one-class classification. Inf Sci 264:182–195
18. Krawczyk B, Wozniak M (2014) Diversity measures for one-class classifier ensembles. Neurocomput 126:36–44
19. Cyganek B (2012) One-class support vector ensembles for image segmentation and classification. J Math Imaging Vis 42(2):103–117
20. Wilk T, Wozniak M (2012) Soft computing methods applied to combination of one-class classifiers. Neurocomput 75(1):185–193
21. Rahmanimanesh M, Jalili S, Sharafat AR (2013) Fusion of one-class classifiers for protocol-based anomaly detection in aodv-based mobile ad hoc networks. Int J Ad Hoc Ubiquitous Comput 14(3):158–171
22. Tax DMJ, Juszczak P (2003) Kernel whitening for one-class classification. Int J Pattern Recognit Artif Intell 17(3):333–347
23. Guo SM, Chen LC, Tsai JSH (2009) A boundary method for outlier detection based on support vector domain description. Pattern Recognit 42:77–83
24. Lee K, Kim D, Lee KH, Lee D (2007) Density-induced support vector data description. IEEE Trans Neural Netw 18(1):284–289
25. Yager RR (1988) On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans Syst Man Cybern 18:183–190
26. Merigo JM, Gil-Lafuente AM (2011) Fuzzy induced generalized aggregation operators and its application in multi-person decision making. Expert Syst Appl 38:9761–9772
27. Merigo JM, Casanovas M (2011) Decision-making with distance measures and induced aggregation operators. Comput Ind Eng 60:66–76
28. Huang HP, Liu YH (2002) Fuzzy support vector machines for pattern recognition and data mining. Int J Fuzzy Syst 4(3):826–835
29. Lin CF, Wang SD (2002) Fuzzy support vector machines. IEEE Trans Neural Netw 13(2):464–471
30. Zhang Y, Chi ZX, Li KQ (2009) Fuzzy multi-class classifier based on support vector data description and improved pcm. Expert Syst Appl 36:8714–8718
31. Forghani Y, Yazdi HS, Effati S (2011) An extension to fuzzy support vector data description (fsvdd*). Pattern Anal Appl. doi:10.1007/s10044-011-0208-z
32. Liu B, Xiao Y, Cao L, Hao Z, Deng F (2013) Svdd-based outlier detection on uncertain data. Knowl Inf Syst 34(3):597–618
33. Huang G, Chen H, Zhou Z, Yin F, Guo K (2011) Two-class support vector data description. Pattern Recognit 44:320–329
34. GhasemiGol M, Monsefi R, Yazdi HS (2010) Intrusion detection by ellipsoid boundary. J Netw Syst Manag 18:265–282
35. Peng X, Xu D (2012) Efficient support vector data descriptions for novelty detection. Neural Comput Appl 21(8):2023–2032
36. Liu YH, Liu YC, Chen YJ (2010) Fast support vector data description for novelty detection. IEEE Trans Neural Netw 21(8):1296–1313

https://freepaper.me/t/۳۸۸۴۳۸ : ترجمه خودت ترجمیار

37. Perkins C, Royer E (1999) Ad hoc on demand distance vector routing. In: Proceedings of second IEEE Workshop on mobile computing systems and applications (WMCSA 99), pp 90–100
38. Perkins C, Belding-Royer E, Das S (2003) Ad hoc on demand distance vector routing. IETF RFC 3561
39. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27:1–27:27. http://www.csie.ntu.edu.tw/~cjlin/libsvm
40. (2010) Ns-2 (network simulator version 2.34). http://www.isi.edu/nsnam/ns/ns-documentation
41. Bache K, Lichman M (2013) Uci machine learning repository. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml

https://freepaper.me/t/۳۸۸۴۳۸ : خودت ترجمه کن مترجمیار