J Netw Syst Manage (2010) 18:265–282
DOI 10.1007/s10922-010-9165-x

Intrusion Detection by Ellipsoid Boundary

Mohammad GhasemiGol · Reza Monsefi · Hadi Sadoghi-Yazdi

Published online: 1 May 2010

Abstract This paper presents a novel approach to describe the normal behavior of computer networks (as used in IDS) based on Support Vector Data Description (SVDD). In the proposed method we find a minimal hyper-ellipse around the normal objects in the input space. Hyper-ellipse can be expanded in high dimensional space (ESVDD) or to be used as a preprocessing in SVDD method (PESVDD) to obtain better results for IDS. KDD-cup99 has been used as data set for test of the proposed method. The overall experiments show prominence of our work in comparison with similar previous works.

Keywords Ellipsoid boundary · Intrusion detection system (IDS) · Support vector data description (SVDD)

1 Introduction

Intrusion detection is one major research problem in network security which analyzes the network traffic and looks for potential threats. There are two general approaches to intrusion detection: Misuse Intrusion Detection (MID) and Anomaly Intrusion Detection (AID) [1–3]. Misuse detection systems, detect known attacks using predefined attack patterns and signatures [4, 5]. Anomaly detection systems detect attacks by observing deviations from the normal behavior of the system [6].

M. GhasemiGol (✉) · R. Monsefi · H. Sadoghi-Yazdi
Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Mashhad, Iran
e-mail: ghasemigol@wali.um.ac.ir; m.ghasemigol@gmail.com

R. Monsefi
e-mail: monsefi@um.ac.ir

H. Sadoghi-Yazdi
e-mail: h-sadoghi@um.ac.ir

Both groups of methods have their advantages and disadvantages. Misuse detection methods are fundamentally limited to known attacks. On the other hand, anomaly detection methods are capable to detect known and unknown attacks. However, new legitimate behavior can be falsely identified as an attack, resulting in a false positive. The problem with current state-of-the-art is to reduce false negative and false positive rate, although it is difficult to achieve both simultaneously. The capability of detecting new attacks makes anomaly detection methods an interesting topic of active research.

The one-class classification problem is a kind of anomaly detection systems among machine learning techniques. In this kind of classification, we assume one class of data as target class and the rest classified as outlier. One-class classification is particularly significant in applications where only a single class of data objects is applicable and easy to obtain. Objects from other classes could be too difficult or expensive to be made available. So we would only describe the target class to separate it from outlier.

Three general approaches have been proposed to resolve the one-class classification problems [7] as follows:-

(1) The most straightforward method to obtain a one-class classifier is to estimate the density of the training data and to set a threshold on its density. Several distributions can be assumed, such as a Gaussian or a Poisson distribution. The most popular three density models are Gaussian model, mixture of Gaussians and Parzen density [8, 9].
(2) In second method a closed boundary around the target set is optimized. K-centers, nearest neighborhood method and support vector data description (SVDD) are example of boundary methods [10, 11].
(3) Reconstruction methods are other one-class classification methods which have not been primarily constructed for one-class classification, but rather to model data. By using prior knowledge about data and making assumptions about the generating process, a model is chosen and fitted to data. Some types of reconstruction methods are: the k-means clustering, learning vector quantization, self-organizing maps, PCA, a mixture of PCAs, diabolo networks, and auto-encoder networks.

The SVDD is a kind of one-class classification method based on Support Vector Machine [12]. It tries to construct a boundary around the target data by enclosing the target data within a minimum hyper-sphere. Inspired by the support vector machines (SVMs), SVDD decision boundary is described by a few target objects, known as support vectors (SVs). One more flexible boundary can be obtained with introduction of kernel functions, by which data are mapped into a high-dimensional space. The most commonly used kernel function is Gaussian kernel [13].

Nowadays, the SVDD method widely uses in outlier detection problems [14–18]. Intrusion detection is a big and complex outlier detection problem. In this problem normal behaviors are always accessible contrary to intrusion ones. Obtaining the intrusion samples is very costly process. Hence we can use SVDD and other one-class classification as solution. These methods just focus on normal behaviors and do not need the intrusion patterns to generate decision boundaries.

In this paper we propose a novel one-class classifier based on the SVDD method which is applied in intrusion detection. In this method we try to fit a better boundary around the normal samples. The SVDD method sometimes could not obtain an appropriate decision boundary in the input space and the good result depends on the proper kernel function. Generally, it is been found that by describing the target class with more precision in the input space; the better results will be obtained. Hence, we define a hyper-ellipse around the target class to get a tighter boundary. Indisputably hyper-ellipse constructs a better decision boundary from hyper-sphere in the input space. The experiments confirm the prominence of our proposed method against the standard SVDD in both input and high-dimensional space. Furthermore we use this hyper-ellipse to define a mapping function as a preprocessing for the SVDD method for intrusion detection.

The paper is organized as follows. In the next section intrusion detection methods are reviewed. In Sect. 3 the SVDD method is described. The proposed methods are explained in Sect. 4, and eventually, experimental results are presented.

2 Intrusion Detection Methods

Network security aims to protect the entire infrastructure of a computer network and its corresponding services from unauthorized access. There are several fundamental components in network security:-

(1) Security-specific infrastructures, such as hardware- and software-based firewalls and physical security approaches.
(2) Security polices, which include security protocols, users' authentications, authorizations, access controls, information integrity and confidentiality.
(3) Detection of malicious programs, including anti-viruses, worms, or Trojan horses, and spyware or malware.
(4) Intrusion detection and prevention, which encompasses network traffic surveillance and analyzing and profiling user behavior [19].

Since James P. Anderson outlined the increasing awareness of computer security problems and presented a project plan to address computer security challenges in 1972 for the United States Air Force (USAF), interest in intrusion detection research has been growing. For more than three decades, it widely used to overcome security threats in computer networks and to identify unauthorized use, misuse, and abuse of computer systems.

In general, intrusion detection is the process of identifying and responding to malicious activity targeted at computing and networking resources [20]. According to this definition an intrusion detection system (IDS) is a security management system for computers and networks which gathers and analyzes data from various areas within a computer or a network to identify possible security breaches. IDS are split into two groups, anomaly detection systems and misuse detection systems.

Each of these approaches has its own strengths and weaknesses. Misuse detection systems generally have very low false positive rates, which indicate error rates of mistakenly detected non-intrusion cases. For this reason, this approach can be seen

at work in the majority of commercial systems [21]. However, they are unable to identify novel or obfuscated attacks, leading to high false negative rates, which represent error rates of missed detection cases.

Anomaly-based systems, on the other hand, are able to detect novel attacks but currently produce a large number of false positives. This stems from the inability of current anomaly-based techniques to cope adequately with the fact that in the real world normal, legitimate computer network and system usage changes over time, meaning that any profile of normal behavior also needs to be dynamic and has to be kept updated [22].

There are many various techniques which have been utilized for intrusion detection such as statistical approaches, data mining, Fourier analysis, hybrid model, agent-based approach and kernel approach. Data mining is an information extraction approach used to discover hidden facts contained in data. It employs a combination of machine learning, statistical analysis, modeling techniques, and database technology to find patterns and subtle relationships in data and infers rules that allow the prediction of future results. We can apply data mining to develop rules that accurately capture the behavior of intrusions and normal activities. Three types of algorithms that could be particularly useful for mining audit data:

(1) Classification: to map a data item into one of several predefined categories,
(2) Link analysis: to determine relationships between fields in database records, and
(3) Sequence analysis: to discover what time-based sequences of audit events frequently occur concurrently.

The data mining based detection models performed as good as the best system built using manual knowledge engineering approaches.

Here we focus on data mining approaches because we can easily look at the problem of intrusion detection as a data clustering issue. The main goal in intrusion detection systems is the separation of normal samples from the outliers. Hence one-class classification methods can be useful for this purpose. These classification methods can describe target objects more accurately. In the following subsection we explain one of the best methods in description of data in special domain.

3 Support Vector Data Description (SVDD)

The SVDD is a one-class classification method that estimates the distributional support of a dataset. A flexible closed boundary function is used to separate trustworthy data on the inside from outliers on the outside [12, 23].

The basic idea of SVDD is to find a minimum hyper-sphere containing all the objective samples and none of the nonobjective samples. The hyper-sphere is specified by its center a and its radius R. The data description is achieved by minimizing the error function:

$$F(R, a) = R^2 \tag{1}$$

$$s.t. \quad \|x_i - a\|^2 \leq R^2, \quad \forall i. \tag{2}$$

In order to allow for outliers in the training dataset, the distance of each training sample x_i to the center of the sphere should not be strictly smaller than R^2. However, large distances should be penalized. Therefore, after introducing slack variables $\xi_i \geq 0$ the minimization problem becomes:

$$F(R, a) = R^2 + C \sum_i \xi_i, \tag{3}$$

$$s.t. \quad \|x_i - a\|^2 \leq R^2 + \xi_i, \quad \forall i. \tag{4}$$

The parameter C gives the tradeoff between the volume of the description and the errors. The constraints can be incorporated into the error function by introducing Lagrange multipliers and constructing the Lagrangian.

$$L(R, a, \alpha_i, \gamma_i, \xi_i) = R^2 + C \sum_i \xi_i - \sum_i \alpha_i \left\{ R^2 + \xi_i - \left(\|x_i\|^2 - 2a.x_i + \|a\|^2 \right) \right\} - \sum_i \gamma_i \xi_i \tag{5}$$

With the Lagrange multipliers of $\alpha_i \geq 0$, and $\gamma_i \geq 0$. Setting partial derivatives to 0 gives these constraints:

$$\frac{\partial L}{\partial R} = 0 : \quad \sum_i \alpha_i = 1 \tag{6}$$

$$\frac{\partial L}{\partial a} = 0 : \quad a = \frac{\sum_i \alpha_i x_i}{\sum_i \alpha_i} = \sum_i \alpha_i x_i, \tag{7}$$

$$\frac{\partial L}{\partial \xi_i} = 0 : \quad C - \alpha_i - \gamma_i = 0 \tag{8}$$

From the above equations and the fact that the Lagrange multipliers are not all negative, when we add the condition $0 \leq \alpha_i \leq C$. Lagrange multipliers γ_i can be safely removed. So the problem can be transformed into maximizing the following function L with respect to the Lagrange multipliers α_i:

$$L = \sum_i \alpha_i (x_i.x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i.x_j), \tag{9}$$

$$s.t. \quad 0 \leq \alpha_i \leq C. \tag{10}$$

Note that from (7), the center of the sphere is a linear combination of the training samples. Only those training samples x_i which satisfy (4) by equality are needed to generate the description since their coefficients are not zero. Hence, these samples are called Support Vectors. The radius can be computed using any of the support vectors:-

$$R^2 = (x_k.x_k) + 2 \sum_i \alpha_i (x_i.x_k) - \sum_{i,j} \alpha_i \alpha_j (x_i.x_j) \tag{11}$$

To judge a test sample z whether is in the target class, its distance to the center of sphere is computed and compared with R. It will be accepted, if satisfies (12), and rejected otherwise.

$$\|z-a\|^2 = (z.z) + 2\sum_i \alpha_i (z.x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i.x_j) \leq R^2. \tag{12}$$

SVDD is stated in terms of inner products. For more flexible boundaries, therefore, inner products of samples $(x_i \cdot x_j)$ can be replaced by a kernel function $K(x_j, x_j)$, where $K(x_i, x_j)$ satisfies Mercer's theorem [23]. This implicitly, maps samples into a nonlinear space to obtain a more tight and non-linear boundaries. In this context, the SVDD problem of (9) can be expressed as:

$$L = \sum_i \alpha_i K(x_i.x_i) - \sum_{i,j} \alpha_i \alpha_j K(x_i.x_j). \tag{13}$$

Several kernel functions have been proposed for the SV classifier. Not all kernel functions are equally useful for the SVDD. It has been demonstrated that using the Gaussian kernel:

$$K(x,y) = \exp\left(-\frac{\|x-y\|^2}{S^2}\right), \tag{14}$$

results in tighter description. By changing the value of S in the Gaussian kernel, the description transforms from a solid hyper-sphere to a Parzen density estimator.

4 The Proposed Methods

The SVDD method does not seem to get a good decision boundary in the input space. For example, Fig. 1 shows the different datasets in the 2-dimensional space. The SVDD method fits a sphere around the target class as a separator in the input space. In this situation, the generated boundary covers the large amount of outlier space incorrectly. It just generates an acceptable boundary in spherical datasets (Fig. 1b).

We propose a new method which generates a better decision boundary around the target class in the input space. In this space a more precision boundary is obtained if

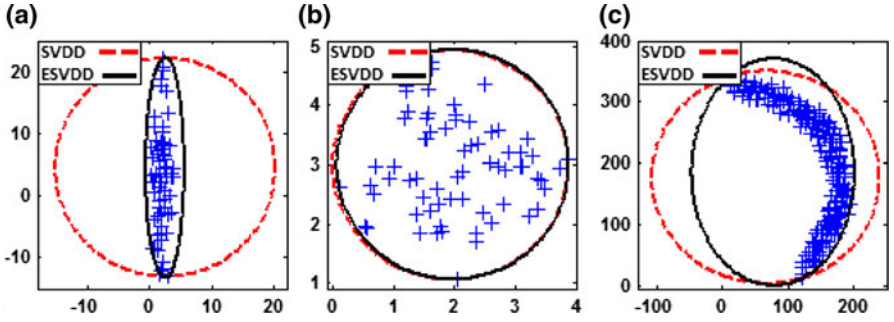

Fig. 1 SVDD and ESVDD boundary in the input space **a** Elliptical dataset **b** Spherical dataset **c** Banana dataset

we use an ellipse instead of a sphere which is presented in the SVDD method. In the high dimensional input space, we can also use a hyper-ellipse as a substitute for a hyper-sphere. Although this technique is more useful for the elliptical datasets (Fig. 1a), it also generates noticeable results on the other datasets such as banana datasets (Fig. 1c). On the other hand an ellipse is a general form of a sphere; so in the worst case it transforms to a sphere and we obtain the same decision boundary as the standard SVDD method (Fig. 1b).

Here we try to find a hyper-ellipse with a minimum volume which encloses all or most of these target objects [26]. This technique results in a better decision boundary in the high-dimensional space as well as the input space. The best results for SVDD method depends on using Gaussian kernel. This kernel transforms input space into a high-dimensional space with infinite dimensions. Indisputably in this space a hyper-ellipse with infinite radii is a more accurate boundary in comparison to a hyper-sphere with just one radius, because these radii help in tightening the boundary of target objects and describing them more precisely.

The problem of finding the minimum hyper-ellipse around n samples with d dimensions represented by a center a and the radii R_j which can be formulated as:

$$F = \sum_j R_j^2, \tag{15}$$

$$s.t. \quad \sum_j \left(\frac{x_{i,j} - a_j}{R_j}\right)^2 \leq 1, \quad \forall i,j. \tag{16}$$

Corresponding to the presented SVDD to allow for outliers in the training dataset, each training sample x_i should not be strictly in the hyper-ellipse. However, large distances should be penalized, so that, after introducing slack variables $\xi_i \geq 0$ the minimization problem becomes:

$$F = \sum_j R_j^2 + C \sum_i \xi_i, \tag{17}$$

$$s.t. \ \sum_j \left(\frac{x_{i,j} - a_j}{R_j}\right)^2 \leq 1 + \xi_i, \ \xi_i \geq 0, \ \forall i,j. \tag{18}$$

where C controls the trade-off between the hyper-ellipse volume and the description error. In order to solve the minimization problem in (17), the constraints of (18) are introduced to the error function using Lagrange multipliers:

$$L(R_j, a_j, \alpha_i, \gamma_i, \xi_i) = \sum_j R_j^2 + C \sum_i \xi_i - \sum_i \alpha_i \left\{ 1 + \xi_i - \sum_j \left(\frac{x_{i,j} - a_j}{R_j}\right)^2 \right\} - \sum_i \gamma_i \xi_i. \tag{19}$$

where $\alpha_i \geq 0$ and $\gamma_i \geq 0$ are Lagrange multipliers. Note that for each object x_i in dimension j a corresponding α_i and γ_i are defined. L has to be minimized with respect to R_j, ξ_i and maximized with respect to α_i and γ_i. A test object z is accepted when it satisfies the following inequality

$$\sum_j \left(\frac{z_{i,j} - a_j}{R_j}\right)^2 \leq 1. \tag{20}$$

In SVDD method we can easily use the kernel trick to get more precise results, because it is stated in terms of inner products. However (18) is very complicated and we cannot apply the kernel trick to it. Hence we extract the transformation functions from the kernel functions and use these functions to create the high dimensional space. This technique is completely explained in the next subsections.

Here we can apply two techniques for getting better results. In the first manner we can generate a high dimensional space and expand this hyper-ellipse in this new space. Secondly we can use the achieved hyper-ellipse as a preprocessing in the SVDD method to obtain better results.

4.1 Fitting a Hyper-Ellipse in the High Dimensional Space (ESVDD)

Similar to SVDD, we can obtain better results by using the hyper-ellipse model. Assume we have a mapping Φ for data fitting. We can apply this mapping to (17) and we obtain:

$$\begin{aligned} L(R_j, a_j, \alpha_i, \gamma_i, \xi_i) = &\sum_j R_j^2 + C\sum_i \xi_i \\ &- \sum_i \alpha_i \left\{1 + \xi_i - \sum_j \left(\frac{\Phi(x_i)_j - a_j}{R_j}\right)^2\right\} - \sum_i \gamma_i \xi_i. \end{aligned} \tag{21}$$

According to [24] we can get these Φ functions from the standard kernels which is been proposed for the support vector classifier. For example to find the Φ function for polynomial kernel with $d = 2$, in the 2-dimensional space, we should do the following procedure:

$$x = [x_1, x_2]$$

$$x = [y_1, y_2]$$

$$k(x, y) = (1 + x^T y)^2$$

$$k(x, y) = 1 + x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 + 2x_1 y_1 + 2x_2 y_2 k(x, y) = \Phi(x)^T \Phi(y)$$

$$\Phi(x) = [1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2] \tag{22}$$

In the SVDD method, using the Gaussian kernel instead of the polynomial kernel results in tighter descriptions. So we use this kernel for a random ellipsoid dataset. First we extract the Φ function from the Gaussian kernel. For this reason we solve the following problem:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{S^2}\right) = \Phi(x)^T \Phi(y) \tag{23}$$

If we suppose $S = 1$ then:

$$
\begin{aligned}
k(x,y) &= e^{-\|x_i - y_i\|}_{i=1,\ldots,n} = e^{-[(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2]} \\
&= e^{-(x_1-y_1)^2} * e^{-(x_2-y_2)^2} * \cdots * e^{-(x_n-y_n)^2} \\
&= e^{-x_1^2} * e^{-y_1^2} * e^{2x_1y_1} * \cdots * e^{-x_n^2} * e^{-y_n^2} * e^{2x_ny_n} \\
&= e^{-(x_1^2+\cdots+x_n^2)} * e^{-(y_1^2+\cdots+y_n^2)} * e^{2x_1y_1} * \cdots * e^{2x_ny_n}
\end{aligned}
$$

The Taylor formula for $e^{2x_iy_i}$ is

$$
e^{2x_iy_i} = 1 + 2x_iy_i + \frac{4x_i^2y_i^2}{2!} + \cdots + \frac{2^n x_i^n y_i^n}{n!} \tag{24}
$$

By using proper substitutions we can get desired Φ function. For example in the 1-dimensional input space and using four terms to compute the Taylor formula, the following Φ function is obtained.

$$
\begin{aligned}
K(x_1,y_1) &= e^{-x_1^2} * e^{-y_1^2} * e^{2x_1y_1} = \Phi(x)^T\Phi(y) \\
&= e^{-x_1^2} * e^{-y_1^2} * \left(1 + 2x_1y_1 + \frac{4x_1^2y_1^2}{2!} + \frac{8x_1^3y_1^3}{3!}\right) \\
\Phi(x) &= [e^{-x_1^2}, \sqrt{2}x_1e^{-x_1^2}, \sqrt{2}x_1^2e^{-x_1^2}, \frac{2}{\sqrt{3}}x_1^3e^{-x_1^2}]
\end{aligned} \tag{25}
$$

For the Gaussian kernel no finite mapping $\Phi(x)$ of object x can be given. But we can get an approximation of Φ functions by using the Taylor formula. So we can use these functions for mapping input space into high-dimensional space.

4.1.1 Φ Functions characteristics

Now we confronted a new difficulty about the Φ functions. As mentioned in the previous section, the Gaussian kernel has no finite mapping $\Phi(x)$ of object x. So we use an approximation of Φ functions. If we consider these functions with more attention, interesting results are obtained.

For example in the 4-dimensional space we get a Φ function which is mapping this space into 10,000-dimensional space, although many of these dimensions contained very small coefficient. So eliminating of these dimensions does not impose a critical error in the final results.

Figure 2 shows the logarithm of coefficients for each dimension. Just few dimensions have considerable coefficients which can be efficient in the transformed domain. Therefore we can map 4-dimensional space into a smaller high-dimensional space with fewer dimensions. These dimensions are selected from 10,000 dimensions which have considerable coefficients (bigger than 10^{-6}). Hence many of these dimensions are useless and can be eliminated.

4.2 Applying a Preprocessing in the SVDD Method (PESVDD)

In this section we proposed a new preprocessing technique for SVDD method which can lead to better results in practice. In the first step, we find the minimal hyper-

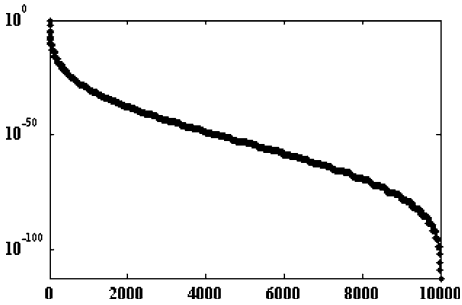

Fig. 2 The logarithm of coefficients for each dimension

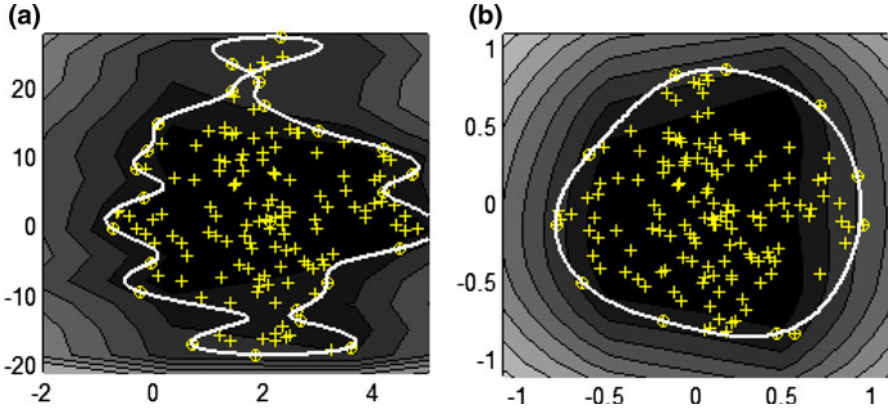

Fig. 3 Decision boundary for an elliptical dataset **a** without applying mapping functions **b** by using mapping functions

ellipse which covers the target objects in the input space. Hence we have to solve this optimization problem which explained before. The achieved minimum hyper-ellipse around n target samples with j dimensions represented by a center a and the radii R_j ($j = 1, \ldots, d$). Then we normalize data in respect to center and radii of obtained hyper-ellipse. In other words we define a simple mapping function which is shown in (26) to decrease the variance of data.

$$X_{\text{new}} = \frac{x - a}{R} \tag{26}$$

According to this transformation the new samples have smaller variance and would be suitable for using in the SVDD method. We show that this preprocessing has two advantages; on the one hand it leads to generate better decision boundary and on the other hand the final results are less influenced by changing the user defined parameters.

Here we apply the SVDD method to a simple synthetic dataset with and without this preprocessing technique. Figure 3 shows that the usage of this mapping function decreases the variance of samples. In this manner we obtain a very smooth and useful decision boundary.

5 Experiments

We compare the performances of the SVDD and new proposed methods with a synthetic dataset and some of datasets taken from UCI Machine Learning Dataset Repository [25]. In the last of this section we apply this method for intrusion detection problem. Table 1 provides details about the datasets that are used here.

In Iris and balance-scale datasets three classes with four features are existed. Since, to use them for outlier detection, two of the classes are used as the target class while the remainder class is considered as outlier. Haberman's Survival dataset has two classes and uses three features. In this dataset we have 225 samples in one class and 81 samples in the other one. So we can use it easily for a one class classification problem. In this situation, the class with more samples supposed as the target class. In the first step we should create some Φ functions for mapping the various input space into a high dimension high-dimensional space.

Here we deal with datasets by 2, 3 or 4 dimensions. The related mapping function characteristics are presented in Table 2. For example Φ_1 is a mapping function which transforms a sample with 2 dimensions into a new position in a 100 dimensions space. We repeat all of the experiments with some of the selected dimensions of Φ functions (Φ').

According to the previous section we claim that reducing the dimensions in the high-dimensional space has not any critical effects in the classification. Even in some cases fewer dimensions lead to better results. Table 3 compares the

Table 1 UCI datasets used for the evaluation of the outlier detection methods

Dataset	No. of objects	No. of classes	No. of features
Iris	150	3	4
Haberman	306	2	3
Balance-scale	625	3	4

Table 2 Mapping functions characteristics (a) Φ functions characteristics (b) Selected dimensions from Φ functions

Φ functions	Input space dimensions	High-dimensional space dimensions
(a)		
Φ_1	2	100
Φ_2	3	1,000
Φ_3	4	10,000
Φ' functions	Input space dimensions	High-dimensional space dimensions
(b)		
Φ'_1	2	6
Φ'_2	3	10
Φ'_3	4	15

Table 3 Recognition Rate in SVDD and ESVDD methods for the Iris, balance-scale and heberman datasets

Experiment conditions	SVDD	Mapping function	ESVDD	PESVDD
Iris dataset	85.3%	Φ_3	89.0%	92.4
(Target class = class 1 & class 2)				
(Outliers = class 3)				
Random learning samples = 50		Φ'_3	87.7%	
Random testing samples = 100				
(50 target + 50 outlier)				
Balance-scale dataset	65.3%	Φ_3	67.4%	68.2
(Target class = class L & class R)				
(Outliers = class B)				
Random learning samples = 40		Φ'_3	66.2%	
Random testing samples = 98				
(49 target + 49 outlier)				
Heberman dataset	69.7%	Φ_2	78.6%	77.5
(Target class = class 1)				
(Outliers = class 2)				
Random learning samples = 40		Φ'_2	78.9%	
Random testing samples = 100				
(50 target + 50 outlier)				
Synthetic dataset	92.6%	Φ_1	96.9%	97.3
Iteration = 100				
(Target class = class 1)				
(Outliers = class 2)				
Learning samples = 150		Φ'_1	97.6%	
Testing samples = 300				
(150 target + 150 outlier)				

performance of SVDD and ESVDD methods with Iris, balance-scale and haberman datasets in 10 Iterations.

The prominence of our proposed methods will be so clear when we use an elliptical synthetic dataset. In these experiments, we use the same user defined parameters. Thus ESVDD generates better decision boundary than SVDD in the non-spherical datasets specially.

The KDD-cup 1999 data was created based on the 1998 MIT-DARPA, it is available to download from the Information and Computer Science website, at the University of California at Irvine. Similar to the 1998 MIT-DARPA data, the full KDD-cup 1999 dataset, includes 7 weeks of TCP dump network traffic training data that was processed into approximately five million connection records, and 2 weeks of testing data, comprised of 38 different attack types that can be grouped into 5 categories which are shown in Table 4.

Table 4 Frequencies of major attacks from KDD-cup 1999 data

Attack types	Training data ($n = 494{,}021$) No.	Testing data ($n = 311{,}029$) No.
Legitimate users (normal)	97,278	60,593
Surveillance and other probing (probe)	4,107	4,166
Denial of service (DoS)	391,458	229,853
Unauthorized access to local super user (root) privileges (U2R)	52	228
Unauthorized access from a remote machine (R2L)	1,126	16,189

The attacks types are grouped into four groups:

1. Probing: is a class of attacks where an attacker scans a network to gather information or find known vulnerabilities; surveillance and other probing, e.g., port scanning.
2. DOS: Denial of Service; is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a machine, e.g., Syn-flood.
3. U2R: User to root exploits are a class of attacks where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system; unauthorized access to local super user (root) privileges, e.g., various "buffer overflow" attacks.
4. R2L: A remote to user (R2L) attack is a class of attacks where an attacker sends packets to a machine over a network, then exploits machine's vulnerability to illegally gain local access as a user; unauthorized access from a remote machine, e.g., guessing password.

The training data is made up of 22 different attacks out of the 39 present in the test data. Table 7 in the Appendix shows the different attack types for both training (known) and the additional attack types included for testing (novel) for the four categories. The known attack types are those present in the training dataset while the novel attacks are the additional attacks in the test datasets not available in the training datasets.

The testing data does not have the same probability distribution as the training data, and includes additional specific attack types. The KDD-cup 1999 data includes 41 variables that extended from the 11 variables of the raw 1998 MIT-DARPA data. These variables can be found in the Appendix (See Table 8).

5.1 Evaluating Measures

In the world of Data Mining and One-class classification methods some well-defined standard measures exist for evaluation. In general, there are four situations in intrusion detecting procedure which are shown in Table 5. According to this table two kinds of error will occur. In the false positive error we detect a normal sample

Table 5 Occurred situations in intrusion detection in computer networks

Actual class			
Normal	Intrusion		
True positive (TP)	False negative (FN)	Normal	Detected class
False positive (FP)	True negative (TN)	Intrusion	

as an intrusion. On the other hand, detecting an intrusion sample as a normal sample, lead to a false negative error. The other two situations present the right detections.

Different measures are defined with the aid of these four situations. Some of these measures are defined below.

$$\text{precision} = \frac{TP}{TP + FP} * 100 \tag{27}$$

$$\text{recall} = \frac{TP}{TP + FN} * 100 \tag{28}$$

$$F - \text{measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{29}$$

$$\text{CA} = \frac{\text{TP} + \text{TN}}{P(p) + P(n)} * 100 \tag{30}$$

$$\text{DR} = \frac{\text{TP}}{P(p)} \tag{31}$$

$$\text{FAR} = \frac{\text{FP}}{P(n)} \tag{32}$$

In these formulations $P(p)$ and $P(n)$ are the total number of positive and negative samples sequentially. Precision is the number of correct positive predictions compared to the total number of positive predictions. Recall is the number of correct positive predictions compared to the total number of positive data. F-measure is a tradeoff between these two measures. Another common measure is Classification Accuracy (CA) which tells you how well a classification algorithm works. DR computes detection rate and FAR uses to compute the false alarm rate.

5.2 Experiments for KDD-cup 1999 Dataset

Here we apply the proposed methods to KDD-cup 1999 dataset. As an optional step we do a feature selection method to decrease the dimensions of samples. This step has many influences in execution time for ESVDD approach specially. Table 6 gives the experimental results of intrusion detection using ESVDD and PESVDD methods. In these experiments we use Gaussian kernel to get better results. All methods are executed in equal conditions by same learning and testing samples.

Table 6 Experimental results for applying SVDD, ESVDD and PESVDD methods to KDD-cup 1999 dataset

Kind of Intrusion	Precision			Recall			F-measure		
	SVDD	ESVDD	PESVDD	SVDD	ESVDD	PESVDD	SVDD	ESVDD	PESVDD
Dos	77.0	78.4	91.9	97.5	98.4	97.9	86.0	87.3	94.8
U2R	88.5	89.5	95.0	90.8	93.2	96.9	89.6	91.3	95.9
R2L	81.7	80.4	83.9	90.4	94.5	99.0	85.8	86.7	90.8
Probe	65.7	66.6	78.0	95.4	95.3	98.6	77.8	78.4	87.1
Kind of Intrusion	CA			DR			FAR		
	SVDD	ESVDD	PESVDD	SVDD	ESVDD	PESVDD	SVDD	ESVDD	PESVDD
Dos	87.5	88.5	95.0	0.77	0.78	0.92	0.23	0.22	0.08
U2R	89.7	91.5	96.0	0.88	0.89	0.95	0.12	0.11	0.05
R2L	86.5	87.8	91.5	0.82	0.80	0.84	0.18	0.20	0.16
Probe	81.2	81.6	88.4	0.66	0.67	0.78	0.34	0.33	0.22

ESVDD and PESVDD are new one-class classification methods. Hence, we just focus on normal samples and use them in the learning step. Then we can detect attacks by observing deviations from the normal samples.

6 Conclusion

In this paper, we propose a new approach to make the SVDD boundary closely fit the contour of the target data. The SVDD method uses a hyper-sphere which cannot be a good decision boundary for the target data, in the input space. So we define a hyper-ellipse instead of a hyper-sphere and resolve the equations by applying this alteration. Experiments show that using a hyper-ellipse leads to better results in the high-dimensional space beside the input space.

Furthermore we can use the achieved minimal hyper-ellipse as a preprocessing step for the SVDD method to get better decision boundary. As an important benefit, it is less influenced by changing the user defined parameters and we even obtained acceptable results with the inappropriate parameters. Experiments demonstrate the prominence of these methods to detect intrusions in KDD-cup 1999 dataset. Contrary to the other classification methods, the proposed methods are just focused on normal behaviors. Due to this advantage new attacks can be easily recognized.

Acknowledgments This work has been partially supported by Iran Telecommunication Research Center (ITRC), Tehran, Iran (Contract No: T/500/1640). This support is gratefully acknowledged.

Appendix

See Tables 7 and 8.

Table 7 Known and novel attacks in KDD-cup 1999 data

DOS	Known	Back, land, Neptune, Pod, smurf, teardrop
	Novel	apache2, udpstorm, processtable, mailbomb
Probe	Known	ipsweep, satan, nmap, portsweep
	Novel	Saint, mscan
R2L	Known	ftp_write, guess_passwd, warezmaster, warezclient, imap, phf, spy, multihop
	Novel	named, xlock, sendmail, xsnoop, worm, snmpgetattack, snmpguess
U2R	Known	rootkit, loadmodule, buffer_overflow, perl
	Novel	xterm, p.s., sqlattack, httptunnel

Table 8 Variables in KDD-cup 1999 data

Basic features of individual TCP connections	
Length (number of seconds) of the connection	Duration
Type of the protocol {ICMP (yes/no), TCP (yes/no), UDP (yes/no)}	protocol_type
Network service on the destination, HTTP (yes/no)	Service
Number of data bytes from source to destination	src_bytes
Number of data bytes from destination to source	dst_bytes
Normal or error status of the connection {REJ (yes/no), S0 (yes/no), SF (yes/no), RSTO or RSTOS0 or RSTR (yes/no)}	Flag
Connection from/to the same host/port (yes/no)	Land
Number of WRONG fragments	Wrong_fragment
Number of urgent packets	Urgent
Content features within a connection suggested by domain knowledge	
Number of HOT indicators	Hot
Number of failed login attempts	num_failed_logins
Login successfully (yes/no)	Logged_ in
Number of COMPROMISED conditions	num_compromised
Root shell s obtained	root_shell
SU ROOT command attempted (yes/no)	su_attempted
Number of ROOT accesses	num_root
Number of file creation operations	num_file_creations
Number of shell prompts	num_shells
Number of operations on access control files	num_access_files
Number of outbound commands in an ftp session	num_outbound_cmds
HOT login (yes/no)	is_hot_login
GUEST login (yes/no)	is_guest_login
Traffic features computed using a 2-s time window	
Number of connections to the same host as the current connection in the past 2 s	Count
Percent of connections that have SYN errors	Serror_rate
Percent of connections that have REJ errors	Rerror_rate
Percent of connections to the same service	same_srv_rate
Percent of connections to different services	diff_srv_rate

Table 8 continued

Number of connections to the same service as the current connection in the past 2 s	srv_count
Percent of connections that have SYN errors	srv_serror_rate
Percent of connections that have REJ errors	srv_rerror_rate
Percent of connections to different hosts	srv_diff_host_rate
Destination	
Number of connections having the same destination host	dst_host_count
Number of connections having the same destination host and using the same service	dst_host_srv_count
Percent of connections having the same destination host and using the same service	dst_host_same_srv_rate
Percent of different services on the current host	dst_host_diff_srv_rate
Percent of connections to the current host having the same source port	dst_host_same_src_port_rate
Percent of connections to the same service coming from different hosts	dst_host_srv_diff_host_rate
Percent of connections to the current host that have an S0 error	dst_host_serror_rate
Percent of connections to the current host and specified service that have an S0 error	dst_host_srv_serror_rate
Percent of connections to the current host that have an RST error	dst_host_rerror_rate
Percent of connections to the current host and specified service that have an RST error	dst_host_srv_rerror_rate

References

1. Ghosh, K.A., Schwartzbard, A.: Study in using neural networks for anomaly and misuse detection. In: Proceedings of the 8th SENIX security symposium, pp. 131–142. Washington, DC, 23–26 August 1999
2. Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. VLDB J. **16**, 507–521 (2007)
3. Zheng, J., Hu, M.: Intrusion detection of DoS/DDoS and probing attacks for web services. In: Proceedings of the WAIM, pp. 333–344. Hangzhou, China, LNCS, 3739, 11–13 Oct 2005
4. Ilgun, K., Kemmerer, R.A., Porras, P.A.: State transition analysis: a rule-based intrusion detection approach. IEEE Trans. Software Eng. **21**(3), 181–199 (1995)
5. Marchette, D.: A statistical method for profiling network traffic. In: Proceedings of the first USENIX workshop on intrusion detection and network monitoring, pp. 119–128. Santa Clara, California, USA, 9–12 April 1999
6. Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection: support vector machines and neural networks. In: Proceedings of the IEEE international joint conference on neural networks (ANNIE), pp. 1702–1707. St. Louis (2002)
7. Tax, D.M.J.: One-Class Classification: Concept Learning in the Absence of Counter-Examples. Technische Universiteit Delft, Netherlands (2001)
8. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. **33**, 1065–1076 (1962)
9. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Walton Street, Oxford OX2 6DP (1995)
10. Ypma, A., Duin, R.P.W.: Support objects for domain approximation. In: Proceedings of the 8th international conference on artificial neural networks (ICANN'98), Skovde, Sweden, pp. 719–724. Springer, Berlin, 2–4 Sept 1998
11. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Mach. Learn. **54**, 45–66 (2004)
12. Tax, D.M.J., Duin, R.P.W.: Support vector domain description. Pattern Recognit. Lett. **20**, 1191–1199 (1999)

13. Guo, S.M., Chen, L.C., Tsai, J.S.H.: A boundary method for outlier detection based on support vector domain description. Pattern Recognit. **42**, 77–83 (2009)
14. Liu, Y., Gururajan, S., Cukic, B., Menzies, T., Napolitano, M.: Validating an online adaptive system using SVDD. In: Proceedings of the 15th IEEE international conference on tools with artificial intelligence (ICTAI'03), pp. 384–388. Sacramento, California, USA, 3–5 Nov 2003
15. Ji, R., Liu, D., Wu, M., Liu, J.: The application of SVDD in gene expression data clustering. In: Proceedings of the 2nd international conference on bioinformatics and biomedical engineering (ICBBE'08), pp. 371–374. Shanghai, China, 16–18 May 2008
16. Yu, X., Dementhon, D., Doermann, D.: Support vector data description for image categorization from internet images. In: Proceedings of the 19th international conference on pattern recognition (ICPR'08), Tampa, Florida, USA, 8–11 Dec 2008
17. Cho, H.W.: Data description and noise filtering based detection with its application and performance comparison. Expert Syst. Appl. **36**, 434–441 (2009)
18. Jiaomin,L., Zhenzhou,W., Xinchun, F., Jing, W.: Intrusion detection technology based on SVDD. In: Proceedings of the 2nd international conference on intelligent networks and intelligent systems (ICINIS'09), Tianjin, Chaina, 1–3 Nov 2009
19. Wang, Y.: Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection. Inf Sci Ref, Hershey, New York (2009)
20. Amoroso, E.: Intrusion detection: an introduction to internet surveillance, correlation, trace back, traps, and response, 1st edn. Intrusion NetBooks (1999)
21. Kim, J., Bentley, P.J., Aickelin, U., Greensmith, J., Tedesco, G., Twycross, J.: Immune system approaches to intrusion detection—A review. Nat. Comput. Int. J. **6**(4), 413–466 (2007)
22. Northcutt, S., Novak, J.: Network Intrusion Detection. New Riders, 3rd edn. (2003)
23. Scholkopf, B., Smola, A.J., Muller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**, 1299–1319 (1999)
24. Haykin, S.: Nerual Networks a Comprehensive Foundation. Prentic Hall (1999)
25. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. Department of Information and Computer Sciences, University of California, Irvine, Available at http://www.ics.uci.edu/~mlearn/MLRepository.html
26. GhasemiGol, M., Monsefi, R., Sadoghi-Yazdi, H.: Ellipse Support Vector Data Description. EANN 2009, Springer, CCIS 43, pp. 257–268 (2009)

Author Biographies

Mohammad GhasemiGol was born in Birjand, Iran, in 1984. He received the B.S. degree in Computer Engineering from Payame Noor University (PNU), Birjand, Iran, in 2006. He also received the Master's degree in Computer Engineering at Ferdowsi University of Mashhad (FUM), Mashhad, Iran, in 2009. He is a member of Young Iranian Elites Association and Pattern Recognition Group of FUM. His research interests include pattern recognition, intelligent data mining, intrusion detection, machine learning, optimization problems, grid computing, parallel algorithms and parallel programming.

Reza Monsefi graduated in Electronic & Electrical Engineering (Honors' degree) year 1978 from Manchester University, Manchester, U.K. M.Sc in Control engineering, year 1981, from Salford University, Manchester, U.K. Ph.D in Data Communication and Supervisory Control, year 1983, Salford University, Manchester, U.K. The first working Experience as a Research Fellow from 1985 to 1991, was at the Open University, Milton Keynes, U.K And Since 1991 till now as a lecturer at Ferdowsi University of Mashhad, Mashhad, Iran. Current activities include research on Computer Networks, Machine Learning and soft computing, Data and Resource Grid and Machine Vision and image processing.

Hadi Sadoghi-Yazdi received the B.S. degree in Electrical Engineering from Ferdowsi University of Mashhad, Iran, in 1994, and then he received to the M.S. and PhD degrees in Electrical Engineering from Tarbiat Modarres University of Tehran, in 1996 and 2005, respectively. He works in Computer Engineering Department as an assistant professor at Ferdowsi University of Mashhad, Iran. His research interests include optimization, adaptive filtering, image and video processing. He has more than 140 journal and conference publications in subject of interesting area.