Anomaly Intrusion Detection Based on Hyper-ellipsoid in the Kernel Feature Space

Hansung Lee[1], Daesung Moon[1], Ikkyun Kim[1], Hoseok Jung[2] and Daihee Park[3]
[1] SW·Content Research Laboratory, ETRI
Daejeon, Rep. of Korea
[e-mail: mohan@korea.ac.kr, {daesung, ikkim21}@etri.re.kr]
[2] Computer and Information Section, KRIBB
Daejeon, Rep. of Korea
[e-mail: hsjeong@kribb.re.kr]
[3] Dept. of Computer and Information Science, Korea University
Sejong-city, Rep. of Korea
[e-mail: dhpark@korea.ac.kr]
*Corresponding author: Daihee Park

Received July 13, 2014; revised January 18, 2015; accepted February 27, 2015;
published March 31, 2015

Abstract

The Support Vector Data Description (SVDD) has achieved great success in anomaly detection, directly finding the optimal ball with a minimal radius and center, which contains most of the target data. The SVDD has some limited classification capability, because the hyper-sphere, even in feature space, can express only a limited region of the target class. This paper presents an anomaly detection algorithm for mitigating the limitations of the conventional SVDD by finding the minimum volume enclosing ellipsoid in the feature space. To evaluate the performance of the proposed approach, we tested it with intrusion detection applications. Experimental results show the prominence of the proposed approach for anomaly detection compared with the standard SVDD.

Keywords: Anomaly detection, intrusion detection, kernel principal component analysis, minimum enclosing ellipsoid

This work was supported by the ICT R&D program of MSIP/IITP. [13-921-06-002, Cyber targeted attack recognition and trace-back technology based-on long-term historic analysis of multi-source data].

http://dx.doi.org/10.3837/tiis.2015.03.019 **ISSN : 1976-7277**

1. Introduction

Anomaly detection has received significant attention over the past decades, and it spans numerous disciplines and application domains, such as intrusion detection, fraud detection, medical and public health anomaly detection, industrial damage detection, image processing, and sensor networks. In general, different application domains have different definitions of an anomaly, and it is difficult to distinguish anomalies from noise, insofar as noise tends to be similar to anomalies. With intrusion and malware detection in particular, adversaries often try to make malicious behavior appear normal. Hence, it is difficult to detect anomalies [1] . Despite considerable research efforts, anomaly detection remains a difficult and challenging problem.

Intrusion detection is the art of finding suspicious and malicious (i.e., unauthorized, inappropriate, and anomalous) activity on computer and network systems. Intrusion detection can be divided into two major paradigms in terms of a general strategy for detection [2, 3] : misuse intrusion detection (MID) and anomaly intrusion detection (AID). The MID model identifies malicious behavior by pattern-matching based on known patterns of malicious behavior. This is referred to as rule- or signature-based detection, and involves extracting patterns from known and/or observed attacks. The MID model may perform poorly against new types of attack that are unknown to the intrusion detection system, and the signature must be manually updated whenever novel attacks are reported. The AID model, on the other hand, constructs a normal behavior profile and examines deviations from that profile to detect suspicious behavior. The system discriminates attack (i.e., abnormal) behavior by thresholding the value of deviations. This can be considered as a type of novelty detection, and it can be used to detect unobserved (i.e., newly emerging) attacks. The significant basic assumption of AID is that abnormal behavior from the intruder and malware will be evidently distinguishable from the normal behavior of legitimate users and software [3] . As mentioned before, malicious behavior has become more similar to normal behavior and is thus more difficult to detect.

With the significant success of support vector learning methods in intelligent systems, there is on-going research into applying a support vector machine (SVM) to anomaly detection [4-5] . With anomaly detection, one class of data is regarded as the target class, and the remaining data is classified as an outlier (or anomalous data). Because the other class of data might be available only with difficulty, and because only the normal class of data is easy to obtain in general, one-class classification methods are recently adopted for anomaly detection [6-11] . One of the best-known support vector learning methods for anomaly detection (i.e., one-class SVM) is the support vector data description (SVDD) [12] . The SVDD employs a ball for expressing the region of the target class. It is known as an optimization problem for finding a minimum volume enclosing hyper-sphere with a minimal radius R and center a, containing most of the target data. Because the hyper-sphere in the input space can express only a limited region of the target class, the SVDD enhances its expressing capability by using balls defined in the kernel feature space with kernel tricks. However, even with balls in the feature space, the expressing capability of the SVDD can be limited [10, 11, 13-16] . To address the difficulty with standard SVDD, modified SVDDs that find the ellipsoidal decision boundary for normal data are proposed. GhasemiGol *et al*. [14] proposed a modified SVDD, viz., the Ellipse SVDD (ESVDD), and implemented it with AID applications [10-11] . The ESVDD finds the tighter decision boundary of a target class by using the hyper-ellipse around the target class defined in the input space. However, it has limitations in employing various kernel functions for feature space mapping. Wang *et al*. [15] presented the one-class SVM based on the hyper-ellipsoid model, but it requires the solution of computationally expensive second-order cone programming techniques [16]. Rajasegarar *et al*. [16] presented the centered hyper-ellipsoidal support vector machine (CESVM) for anomaly detection in sensor networks, addressing the computational challenge in [15]. The CESVM is a nonparametric anomaly detection model. That is, the training phase and testing phase are not explicitly distinct. In many real intrusion detection applications, the AID systems are trained using the normal dataset in advance. Only then can they be used detect an intrusion or attack. Therefore, it is difficult to implement it directly with conventional AID applications.

In this paper, we propose a new anomaly detection algorithm based on a minimum volume enclosing ellipsoid (MVEE) with kernel principal component analysis (K-PCA), for mitigating the aforementioned limitations of the conventional SVDD by using the ellipsoid defined in the feature space. To evaluate the proposed approach, we conducted experiments with an intrusion detection benchmark dataset, containing abnormal data significantly similar to normal data. Experimental results show that the proposed approach leads to a significant improvement in anomaly detection with the intrusion detection datasets over the standard SVDD.

The remaining parts of this paper are organized as follows. We summarize the previous work related to our study in Section 2. In Section 3, the proposed anomaly detection approach based on MVEE with K-PCA is provided. Experimental results and discussion are provided in Section 4. Finally, some concluding remarks are given in Section 5.

2. Related Work

The main concern of this study is to provide an anomaly detection approach for intrusion detection, one that alleviates the aforementioned limitations of the standard SVDD. This section presents previous research related to AID and summarizes the SVDD.

2.1 Anomaly Intrusion Detection

AID refers to the detection of malicious behavior or intrusion activity in a computer host and network system based on a pre-defined profile of normal behavior. Anomaly detection techniques are applicable in the field of intrusion detection with the basic assumption that intrusion activity is noticeably different from normal system behavior [1, 3] . There are three main categories of AID techniques [17] : statistical-based, knowledge-based, and machine-learning-based methods. Statistical-based models capture the activity of the system to create a stochastic behavior profile. However, it is difficult to model all behavior using stochastic methods. The knowledge-based model is an expert system, and has many virtues, e.g., robustness, flexibility, and scalability. Generally, a human expert manually constructs the knowledge (rule) base. Hence, building a high-quality knowledge base is a difficult and time-consuming process [17, 18] . Machine-learning approaches such as Bayesian Networks, Markov modes, Neural networks, Fuzzy logic, genetic algorithms, clustering, and outlier detection have been extensively studied over the past decade to overcome some of the drawbacks to statistical- and knowledge-based AID [17-25] . Recently, there is much ongoing research to apply data mining and machine-learning techniques to AID for designing more intelligent intrusion detection systems [2, 17] . Because support vector learning shows superior performance in pattern classification and function approximation, it has developed into a viable tool for intrusion detection. There are two types of AID, based on the support vector learning approach [2] : standard SVM-based and one-class SVM-based methods. The standard SVM-based method divides training data into normal and abnormal datasets during the training phase and classifies observed activity into normal and abnormal behavior during the testing phase. With this model, one class affects the training result of the other class, owing to the unbalanced volume of normal and abnormal training datasets. This model may be subject to misclassification for newly emerging attacks by creating a decision boundary including the unobserved area [2] . To overcome this problem, one-class SVM (e.g., SVDD) and its variations have been implemented in AID [7-11] . It is possible to find the decision boundary of the normal class for AID because the training result is not affected by data instances from the abnormal class and does not include the unobserved area.

2.2 Support Vector Data Description

The SVDD method approximates an optimal hyper-sphere in the feature space with a minimal radius R and center $\mathbf{a}$, containing most of the target data. It can be derived as follows [2, 12] : Given a target dataset D consisting of d-dimensional n-data points $D=\{x_i \in \mathrm{R}^d\}_{i=1}^n$, the SVDD is defined as a problem for finding a sphere that minimizes its volume, including the target dataset. It is formulated with the following optimization problem:

$$\begin{array}{ll} \min & L_0(R^2,\mathbf{a},\xi)=R^2+C\sum_{i=1}^n \xi_i \\ s.t. & \| x_i-\mathbf{a}\|^2 \le R^2+\xi_i, \xi_i \ge 0, \forall_i . \end{array} \tag{1}$$

where R^2 is the square value of the sphere's radius, and $\mathbf{a}$ is the center of the sphere. ξ_i is the penalty term that indicates how far *i*-th data points x_i deviate from the sphere's boundary, and *C* is the trade-off constant.

By introducing a Lagrange function and a saddle-point condition, we obtain the following dual problem, known as the quadratic programing (QP) optimization problem:

$$\begin{array}{ll} \min_\alpha & \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j < x_i, x_j > -\sum_{i=1}^n \alpha_i < x_i, x_i > \\ s.t. & \sum_{i=1}^n \alpha_i = 1, \alpha_i \in [0,C], \forall_i . \end{array} \tag{2}$$

Once the dual solution α_i is obtained by solving the QP problem (2), we can define the decision function as follows:

$$\begin{aligned} f(x) &= R^2 - \| x_i - a \|^2 \\ &= R^2 - < x, x > -2\sum_{i=1}^n \alpha_i < x_i, x > \\ &+ \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j < x_i, x_j > \ \ge \ 0 \end{aligned} \tag{3}$$

To express a more complex region of the target class, we can define the hyper-sphere in the kernel feature space with the Mercer kernel trick as follows:

$$\begin{aligned} \min_{\alpha} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(x_i,x_j)-\sum_{i=1}^{n}\alpha_i k(x_i,x_i) \\ s.t. \quad & \sum_{i=1}^{n}\alpha_i = 1, \alpha_i \in [0,C], \forall_i . \end{aligned} \tag{4}$$

In this case, the decision function can be summarized as follows:

$$\begin{aligned} f(x) \quad &= R^2 - k(x,x) - 2\sum_{i=1}^{n}\alpha_i k(x_i,x) \\ &+\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j k(x_i,x_j) \\ &\geq 0 \end{aligned} \tag{5}$$

3. Anomaly Detection Based on MVEE and K-PCA

This section presents an anomaly detection algorithm based on MVEE and K-PCA, alleviating the limitations of standard SVDD by using an ellipsoid defined in the feature space. The proposed approach consists of two phases: the training phase for finding the decision boundary of the normal class in the feature space, and the testing phase for detecting anomalies.

3.1 Training Phase

As mentioned in the introduction, the SVDD generally enhances its descriptive power by using a hyper-sphere defined in the feature space. However, there are some limitations that cannot be overcome by simply adopting the kernel feature space. To mitigate this drawback to conventional SVDD, we defined the MVEE in the empirical kernel feature space. To define the hyper-ellipsoid in the feature space, we employed K-PCA, a variation of Principal Component Analysis (PCA) in a kernel feature space. Using integral operator kernel functions, we can efficiently compute principal components in high-dimensional feature spaces, which are related to the input space by some nonlinear map.

K-PCA can be summarized as follows [26-28] : Given a set of *n*-training data points mapped onto a feature space, $\Phi(x) = \{\Phi(x_i) \in \mathrm{F}\}_{i=1}^{n}$, the covariance matrix in a kernel feature space is defined as follows:

$$\mathbf{C}^{\mathbf{\Phi}} = \frac{1}{n}\sum_{j=1}^{n}\Phi(x_j)\Phi(x_j)^{\mathrm{T}} \tag{6}$$

Then, we have

$$\lambda(\Phi(x_k)\cdot\mathbf{V}) = (\Phi(x_k)\cdot\mathbf{C}^{\mathbf{\Phi}}\mathbf{V}), \quad k = 1, 2, \cdots, n. \tag{7}$$

where $\lambda \geq 0$ are the eigenvalues, and $\mathbf{V}$ are eigenvectors, $\mathbf{V} = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$. By defining the $n \times n$ kernel matrix $\mathbf{K}$ as $K_{ij} = (\Phi(x_i) \cdot \Phi(x_j))$, we can obtain following equation:

$$n\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha} \tag{8}$$

where $\boldsymbol{\alpha}$ denotes the column vector with the entries $\alpha_1, \alpha_2, \cdots, \alpha_n$. We solve the eigenvalue problem of (8) for nonzero eigenvalues. The solutions $\boldsymbol{\alpha}^k$ belong to nonzero eigenvalues, and they are normalized by requiring that the corresponding eigenvectors in a feature space be normalized, such that $(\mathbf{V}^k \cdot \mathbf{V}^k) = 1$.

For the principal component extraction, we compute projections of the image for the training data points $\Phi(x)$ onto eigenvectors $\mathbf{V}^k$ in the feature space.

$$x = (\mathbf{V}^k \cdot \Phi(x)) = \sum_{i=1}^{n} \alpha_i^k k(x_i, x) \tag{9}$$

where $k(x, y)$ is the kernel function. In this paper, we use the radial basis function.

$$k(x, y) = \exp(-\frac{\| x - y \|^2}{2\sigma^2}). \tag{10}$$

Next, we approximate the decision boundary with the hyper-ellipsoid in the feature space. The ellipsoid can be expressed as follows:

$$\mathrm{E} = \{x \mid (x - x_c)^{\mathrm{T}} \mathbf{Q}^{-1} (x - x_c) \leq 1\}, \tag{11}$$

where $\mathbf{Q}$ is symmetric and a positive definite, i.e., $\mathbf{Q} = \mathbf{Q}^{\mathrm{T}} > 0$, and $\boldsymbol{x_c} \in \mathbb{R}^{\boldsymbol{d}}$ is the center of the ellipsoid. The matrix $\mathbf{Q}$ determines how far the ellipsoid extends in every direction from x_c. The length of the semi-axes of ellipsoid E is given by $\sqrt{\lambda_i}$, where λ_i are the eigenvalues of matrix $\mathbf{Q}$, and the directions of the semi-axes of ellipsoid E are the eigenvectors of matrix $\mathbf{Q}$. The volume of the ellipsoid is proportional to $\det(\mathbf{Q})^{1/2}$ [29-31] . An example of an ellipsoid in $\mathbb{R}^2$ is given in **Fig. 1**.

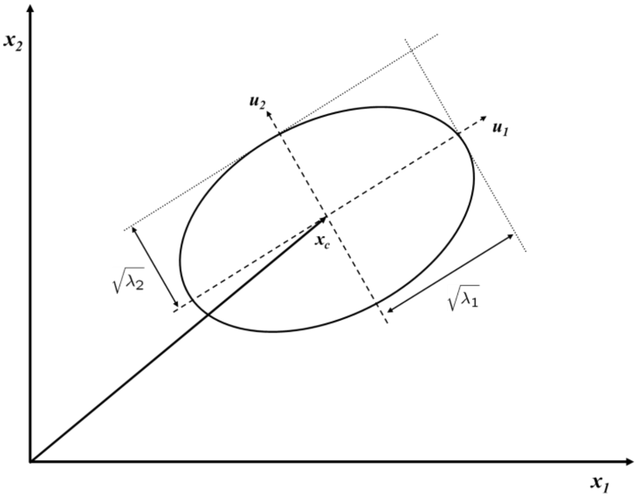


Fig. 1. Example of an ellipsoid in $\mathbf{R}^2$.

Let us consider the ellipsoid approximation problems for computing an MVEE around points in the feature space, $\{x_1, x_2, \cdots, x_n\} \in \mathrm{F}$. This problem is equivalent to finding the minimum volume ellipsoid around the polytope defined by the convex hull of those points. This problem can be reduced to (12) from the ellipsoid expressed by (11).

$$\begin{aligned} \min \quad & \log\det \mathbf{Q} \\ s.t. \quad & \mathbf{Q} = \mathbf{Q}^{\mathrm{T}} > 0, \\ & (x_i - x_c)^{\mathrm{T}} \mathbf{Q}^{-1} (x_i - x_c) \le 1, \\ & i = 1, 2, \cdots, n. \end{aligned} \tag{12}$$

Note that each training data point must be inside the ellipsoidal boundary, and the object function is proportional to the volume of the ellipsoid, represented by the covariance matrix $\mathbf{Q}$. The matrix $\mathbf{Q}$ is a positive-definite matrix, and it is symmetric. By introducing the Schur complement, (12) can be written as the following max-det problem, one of the linear matrix inequality (LMI) problems. The second constraint in (12) can be written as the LMI constraint. However, this problem is difficult to solve directly [29-31] .

$$\begin{aligned} \min \quad & \log\det \mathbf{Q} \\ s.t. \quad & \mathbf{Q} = \mathbf{Q}^{\mathrm{T}} > 0, \\ & \begin{bmatrix} 1 & (x_i - x_c)^{\mathrm{T}} \\ (x_i - x_c) & \mathbf{Q}^{-1} \end{bmatrix} > 0, \\ & i = 1, 2, \cdots, n. \end{aligned} \tag{13}$$

In this paper, we employ MVEE [31] , an efficient and fast approximation algorithm, based on Khachiyan's algorithm [32-34] . The algorithm can be summarized as follows [31] :

Consider a set of n data points in a d-dimensional space, $S = \{x_1, x_2, \cdots, x_n\} \in \mathrm{R}^d$. The minimum volume-enclosing ellipsoid containing S is denoted by MVEE(S). Let a "lifting" of S to R^{d+1} be defined by $S^{'} = \{\pm q_1, \pm q_2, \cdots, \pm q_n\}$, where $q_i^{\mathrm{T}} = [\mathrm{x}_i^{\mathrm{T}}, 1]; \mathrm{i} = 1, 2, \cdots, n$. Based on this definition, each data point x_i is lifted to the hyper-plane, $H = \{(x, x_{d+1}) \in \mathrm{R}^{d+1} \mid x_{d+1} = 1\}$. The MVEE($S^{'}$) is centered at the origin, because $S^{'}$ is centrally symmetric. The minimum volume-enclosing ellipsoid for the original problem is recovered as the intersection of H with the MVEE containing the lifted points q_i , as follows: MVEE(S) = MVEE($S^{'}$) $\bigcap H$. The primal problem with the lifted points q_i can be written as

$$\begin{aligned} \min \quad & \det M^{-1} \\ s.t. \quad & M > 0, \\ & q_i^{\mathrm{T}} M q_i \le 1, \\ & i = 1, 2, \cdots, n. \end{aligned} \tag{14}$$

The Lagrangian dual problem is given by

$$\begin{aligned} \min \quad & \log \det V(u) \\ s.t. \quad & u \ge 0, \\ & 1^{\mathrm{T}} u = 1. \end{aligned} \tag{15}$$

where $V(z) = \mathbf{Q}\mathrm{diag}(\mathrm{z})\mathbf{Q}^{\mathrm{T}}$, $\mathbf{Q} = [q_1, q_2, \cdots q_n]$, and $z = (d+1)u$.

The Lagrangian formulation of (14) can be written as

$$L(M, z) = -\log \det M + \sum_{i=1}^{n} z_i (q_i^{\mathrm{T}} M q_i - 1) \tag{16}$$

By introducing the Karush–Kuhn–Tucker (KKT) conditions for optimality, we obtain the following equality conditions:

$$\begin{aligned} \frac{\partial L}{\partial M} &= -M^{-1} + \sum_{i=1}^{n} z_i q_i q_i^{\mathrm{T}} \\ &= -M^{-1} + \mathbf{QZQ}^{\mathrm{T}} = 0 \end{aligned} \tag{17}$$

where $\mathbf{Z} = \mathrm{diag}(z)$, and $\mathbf{Q} = [q_1, q_2, \cdots, q_n]$. Assume the matrix $M^* > 0$, and $M^* \in \mathrm{R}^{(d+1)\times(d+1)}$ is the optimal solution to the primal problem (14) with the Lagrangian multipliers $z^* \in \mathbb{R}^n$. We then derive

$$V(z^*) = \mathbf{Q}\mathbf{Z}^*\mathbf{Q}^T = (M^*)^{-1} = (d+1)V(u^*) \tag{18}$$

where $V(z) = \mathbf{Q}\mathrm{diag}(z)\mathbf{Q}^{\mathrm{T}}$, and $z = (d+1)u$.

Given $q^{\mathrm{T}} = [x^{\mathrm{T}},1]$, the equation of the ellipsoid can be written as

$$\begin{aligned} \mathrm{MVEE}(S) &= \{x \in \mathrm{R}^d \mid q^T M^* q \leq 1\} \\ &= \{x \in \mathrm{R}^d \mid (\frac{1}{d+1}) q^T V(u^*)^{-1} q \leq 1\}. \end{aligned} \tag{19}$$

Let us denote $\mathbf{Q} = \begin{bmatrix} P \\ 1^{\mathrm{T}} \end{bmatrix} \in \mathrm{R}^{(d+1)\times n}$, where $\mathbf{P} = [q_1, q_2, \cdots, q_n] \in \mathrm{R}^{d\times n}$. We then have

$$\begin{aligned} V(u) = \mathbf{QUQ}^{\mathrm{T}} &= \begin{bmatrix} \mathbf{PUP}^{\mathrm{T}} & \mathbf{P}u \\ (\mathbf{P}u)^{\mathrm{T}} & 1^{\mathrm{T}}u \end{bmatrix} \\ &= \begin{bmatrix} I & \mathbf{P}u \\ 0 & 1 \end{bmatrix} \begin{bmatrix} E^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ (\mathbf{P}u)^T & 1 \end{bmatrix} \end{aligned} \tag{20}$$

where $\mathbf{Q}^{-1} = \mathbf{PUP}^{\mathrm{T}} - \mathbf{P}u(\mathbf{P}u)^{\mathrm{T}}$. The inverse $V(u)^{-1}$ is given by $V(u)^{-1} = \begin{bmatrix} I & 0 \\ -(\mathbf{P}u)^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & -\mathbf{P}u \\ 0 & 1 \end{bmatrix}$. We then have

$$q^T V(u)^{-1} q = (x - \mathbf{P}u)^{\mathrm{T}} \mathbf{Q}(x - \mathbf{P}u) \tag{21}$$

Therefore, for the dual optimal solution u^*, we can obtain

$$\begin{aligned} &\mathrm{MVEE} = \{x \in \mathcal{F} \mid (x - x_c^*)^T \mathbf{Q}^* (x - x_c^*) \leq 1\}, \\ &\mathbf{Q}^* = \frac{1}{d}(\mathbf{PU}^*\mathbf{P}^{\mathrm{T}} - \mathbf{P}u^*(\mathbf{P}u^*)^{\mathrm{T}})^{-1}, \\ &x_c^* = \mathbf{P}u^* \end{aligned} \tag{22}$$

where $P = [q_1, q_2, \cdots, q_n] \in \mathrm{F}$, $q_i^{\mathrm{T}} = [x_i^{\mathrm{T}}, 1]; i = 1, 2, \cdots, n$., u is the dual variable, and $U = \mathrm{diag}(u)$. We obtain the approximated optimal covariance matrix $\mathbf{Q}^*$ and the center x_c^* of the MVEE as the results of the training phase.

3.2 Testing Phase

When the training phase is complete, the test data should be mapped in the same kernel feature space as the training phase in (9), because the decision boundary of the training phase is defined in the kernel feature space. Given a set of *d*-dimensional *m*-testing data points in the input space, $\{x_1^{tst}, x_2^{tst}, \cdots, x_m^{tst}\} \in \mathrm{R}^d$, we can compute projections of the image of the testing data points $\Phi(x^{tst})$ onto eigenvectors $\mathbf{V}^k$ in the kernel feature space. The projection is defined as follows:

$$x^{tst} = (\mathbf{V}^k \cdot \Phi(x^{tst})) = \sum_{i=1}^{n} \alpha_i^k k(x_i^{trn}, x^{tst}) \tag{23}$$

where x^{trn} is a set of n-training data points, and $\mathbf{V}^k$, α^k are obtained in the training phase. An obvious choice for the decision function of the testing data x^{tst} as a target instance is

$$\begin{aligned} f(x^{tst}) &= 1 - (\Phi(x^{tst}) - \overset{*}{x_c})^{\mathrm{T}} \mathbf{Q}^{*} (\Phi(x^{tst}) - \overset{*}{x_c}) \\ &= 1 - (x^{tst} - \overset{*}{x_c})^{\mathrm{T}} \mathbf{Q}^{*} (x^{tst} - \overset{*}{x_c}) \geq 0 \end{aligned} \tag{24}$$

where Q^* is the approximated optimal covariance matrix, and $\overset{*}{x_c}$ is the center of the MVEE, which are obtained from the training phase.

For more flexibility, we modified the decision function (24) ever so slightly as follows [13] :

$$f(x^{tst}) = 1 + \varepsilon - (x^{tst} - \overset{*}{x_c})^{\mathrm{T}} \mathbf{Q}^{*} (x^{tst} - \overset{*}{x_c}) \geq 0 \tag{25}$$

where ε is a constant chosen by the user as a parameter for controlling the boundary of a target class.

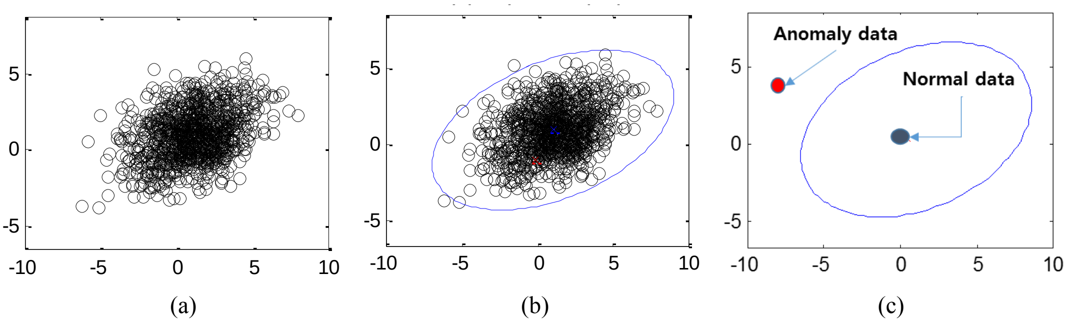


Fig. 2. Illustrations of the training and testing phases for the proposed approach with a two-dimensional toy example: (a) training dataset; (b) decision boundary obtained after the training phase; (c) detection of anomalous data during the testing phase. If a test data point is inside the ellipsoid, the data point is regarded as normal. If a test data point is outside the ellipsoid, the data point is regarded as anomalous.

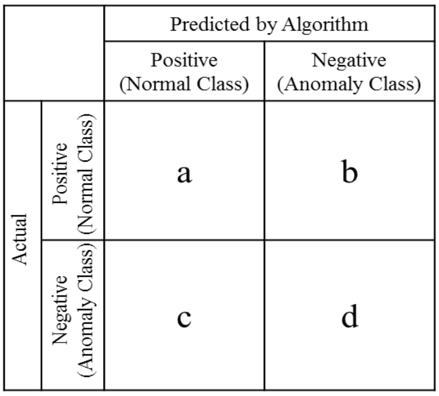
Fig. 2 shows the illustrations of the training and testing phases for the proposed anomaly detection method based on MVEE and K-PCA with a two-dimensional toy example. The training dataset is depicted in **Fig. 2(a)**. The decision boundary for the training dataset that is obtained during the training phase is depicted in **Fig. 2(b)**. During the testing phase, each data point is assigned to one of two classes: the anomalous class or the normal class. If a data point is inside the decision boundary, i.e., inside the ellipsoid, the data point is classified as normal. If a data point is outside the decision boundary, the data point is regarded as anomalous data.

4. Experimental Results and Analysis

This section presents the criteria for evaluating the performance of the proposed approach, along with the experimental results that validate it. In this study, the evaluation measurements were based on a two-class confusion matrix. This evaluation involved determining the decision boundaries of the standard SVDD and the proposed method using two synthetic datasets, in an effort to show the efficacy of the proposed approach. To evaluate the proposed anomaly detection approach, we conducted experiments with intrusion detection datasets, specifically the KDD CUP 1999 dataset [35] and the SNMP MIB traffic-flooding dataset [3] . The experiments described in this paper were conducted on a PC with an Intel(R) Xeon(R) CPU W5590 @ 3.33 GHZ, and all algorithms were implemented using the data description toolbox for Matlab [36] .

4.1 Performance Evaluation Criteria for AID Algorithm

We employed the confusion matrix [37] for the performance evaluations of the proposed AID algorithm. A confusion matrix provides the actual and predicted results obtained by an algorithm. **Fig. 3** shows the confusion matrix for anomaly detection, i.e., a two-class classifier. Here, "actual" refers to the real label of the data points, and "predicted" refers to the label predicted by an algorithm.



		Predicted by Algorithm	
		Positive (Normal Class)	Negative (Anomaly Class)
Actual	Positive (Normal Class)	a	b
	Negative (Anomaly Class)	c	d

Fig. 3. Example of a confusion matrix for the anomaly detection (two-class classification) algorithm. **a**: the number of positive instances correctly classified as a positive class. **b**: the number of positive instances misclassified as a negative class. **c**: the number of negative instances misclassified as a positive class. **d**: the number of negative instances correctly classified as a negative class.

The performance of the algorithm can be evaluated using the data value in the confusion matrix. In this study, we used the following confusion-matrix-based evaluation measurements [11, 37] to evaluate the proposed method. Accuracy (*AC*) is defined as the proportion of the total number of correct predictions.

$$AC = \frac{a+d}{a+b+c+d} \tag{26}$$

The true positive rate (*TPR*) or recall is defined as the proportion of positive instances that are correctly classified as a positive class.

$$TPR = \frac{a}{a+b} \tag{27}$$

The false positive rate (*FPR*) is defined as the proportion of negative instances that are incorrectly classified as a positive class.

$$FPR = \frac{c}{c+d} \tag{28}$$

The true negative rate (*TNR*) is defined as the proportion of negative instances correctly classified as a negative class.

$$TNR = \frac{d}{c+d} \tag{29}$$

The false negative rate (*FNR*) is defined as the proportion of positive instances incorrectly classified as a negative class.

$$FNR = \frac{b}{a+b} \tag{30}$$

Precision (*PRC*) is defined as the proportion of correctly predicted positive (or negative) instances that are predicted as a positive (or negative) class by the algorithm.

$$\begin{aligned} PRC_{Positive} &= \frac{a}{a+c} \\ PRC_{Negative} &= \frac{d}{b+d} \end{aligned} \tag{31}$$

The detection rate (*DR*) is defined as the number of intrusion (attack activity) instances detected by the algorithm divided by the total number of intrusion instances [3] .

$$DR_i = \frac{T_i}{I_i} \tag{32}$$

where I_i is the total number of intrusion instances belonging to an *i*-type attack. T_i is the number of intrusion instances classified as *i*-type of attack by the algorithm.

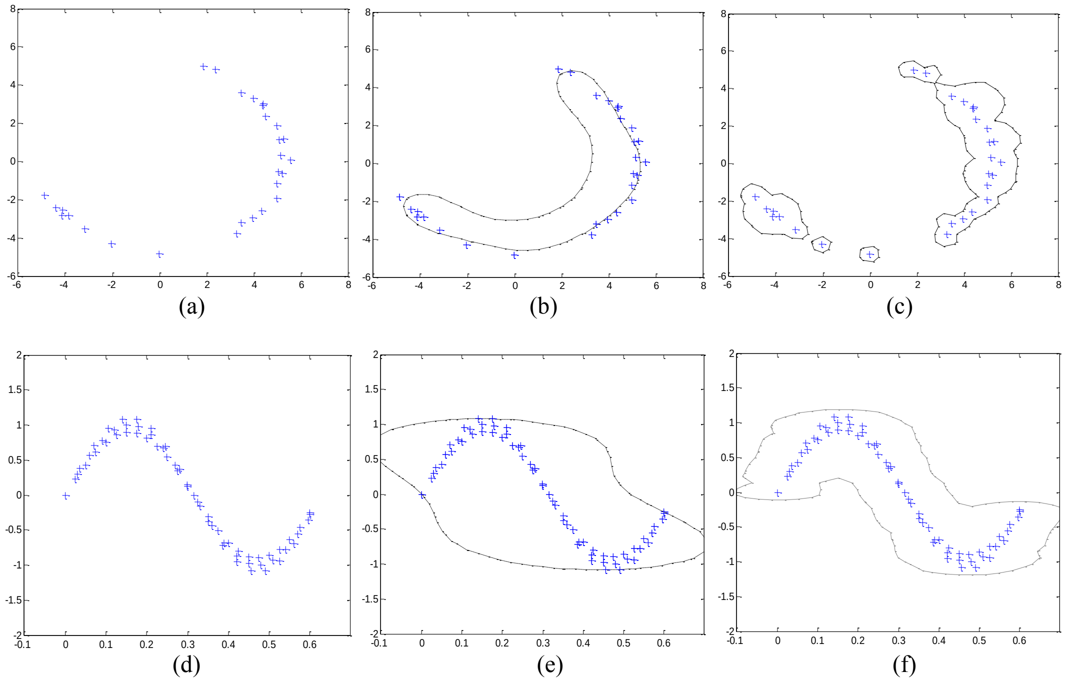
As a single value of merit for comparing different algorithms, we used the *F*-measure [11] , a tradeoff between precision and recall.

$$\begin{aligned} F_{Positive} &= \frac{2\times(PRC_{Positive}\times TP)}{(PRC_{Positive}+TP)} \\ F_{Negative} &= \frac{2\times(PRC_{Negative}\times TP)}{(PRC_{Negative}+TP)} \end{aligned} \tag{33}$$

4.2 Illustrations of Decision Boundaries

To show the efficacy of the proposed approach, we illustrated the decision boundaries of the standard SVDD with a Gaussian kernel function alongside the proposed method, using two synthetic datasets. Dataset 1 is banana shaped, while Dataset 2 is sine-curve shaped. Datasets 1 and 2 are shown in **Fig. 4(a)** and **Fig. 4(d)**, respectively. The decision boundaries of the SVDD (under the conditions σ = 4.0 in Gaussian kernel function, *C*=0.01) and the proposed method (under the condition $2\sigma^2$=0.5 of (10) in K-PCA) with Dataset 1 are depicted in **Fig. 4(b)** and **Fig. 4(c)**, respectively. **Fig. 4(e)** and **Fig. 4(f)** show the decision boundaries for Dataset 2 created by the SVDD (σ = 1.05, *C*=0.01) and the proposed method ($2\sigma^2$=0.01), respectively. As shown in **Fig. 4(b)**, the SVDD finds a compact decision boundary with Dataset 1, which contains most of the data points from the training dataset. The descriptive region for a normal class seems to represent the training dataset well. However, most data instances lie to one side (namely, the bottom-right side) of the decision boundary, and some data points are outside the decision boundary altogether. The description area for Dataset 1 generated by the SVDD is defectively unbalanced between its inner and outer surface. This result suggests a possible defect in SVDD-based anomaly detection approaches. Because the description area for a normal class includes a significantly large area where no training data points reside, it can accept the abnormal data points as a normal class and/or misclassify the normal data points as an abnormal class. Conversely, the proposed method creates a more compact decision boundary with Dataset 1 compared with the standard SVDD, as shown in **Fig. 4(c)**. The decision boundary contains all the data points from the training dataset and the data instances lie inside the decision boundary, almost in the center of the descriptive region. The descriptive area for a normal class includes an area where no training data points reside, an area that is smaller than it is with the SVDD. As noted the introduction, the SVDD can express only a limited region of the target class, even in the feature space.

The experiment results for Dataset 2 are presented in **Fig. 4(e)** and **Fig. 4(f)**. Unlike the experiment with Dataset 1, there is no defective unbalance with Dataset 2 between the area near the inner and outer surfaces of the description region created by the SVDD. However, it still includes a considerably wide area where no training data points reside. The proposed method creates a more compact decision boundary than the SVDD.



(a) (b) (c)

(d) (e) (f)

Fig. 4. Illustrated decision boundaries of the standard SVDD and the proposed method: (a) synthetic Dataset 1, banana shaped; (b) the decision boundary of the SVDD for synthetic Dataset 1, $\sigma = 4.0, C = 0.01$; (c) the decision boundary of the proposed method for synthetic Dataset 1, $2\sigma^2 = 0.5$; (d) synthetic Dataset 2, sine-curve shaped; (e) the decision boundary of the SVDD for Dataset 2, $\sigma = 1.05, C = 0.01$; (f) the decision boundary of the proposed method for Dataset 2, $2\sigma^2 = 0.01$.

As illustrated in **Fig. 4**, the proposed algorithm is able to handle complex shape datasets by using the ellipsoid defined in the feature space. The experiment results on synthetic data show that the proposed algorithm mitigates the limitations of the SVDD and consequently creates a more compact and balanced decision boundary (i.e., description area) compared to the SVDD.

4.3 KDD CUP 1999 Dataset

To determine the performance of the proposed anomaly detection approach, we conducted experiments on one of the best-known benchmark datasets in the field of intrusion detection, namely, the KDD CUP 1999 dataset. The 1998 DARPA Intrusion Detection Evaluation Program collected this dataset during a simulation using U.S. military networks [35] . For accurate analysis of the results, we used only the Corrected-labeled dataset among KDD CUP 1999 datasets. The dataset divides into five classes: normal; denial of service (DOS); unauthorized access from a remote machine (R2L); unauthorized access to local superuser privileges (U2R); and surveillance and other probing (Probe). The dataset consists of 311,029 instances (60,593 normal instances, 229,853 DoS instances, 16,329 R2L instances, 88 U2R instances, and 4,166 Probe instances). The dataset consists of 9 symbolic attributes and 32 numeric attributes. For non-numerical attributes, we borrowed the idea for kernel mapping discrete values of attributes used in [38] . Let Σ_i be the set of possible values of the *i*-th non-numeric attribute and $|\Sigma_i|$ be the cardinality of the set Σ_i, i.e., the number of elements in the set. We encode the values of non-attributes to the $|\Sigma_i|$ coordinates. Each coordinate is mapped to each possible value of the *i*-th non-numerical attribute. The coordinate corresponding to the value of the attribute has a positive value $1/|\Sigma_i|$, and the remaining coordinates have a value of 0. By concatenating numerical attributes and encoded coordinates of non-numerical attributes, we obtained new feature vectors for the SVDD and the proposed algorithm, consisting of only numeric attributes.

For comparison, we set the proposed method against the standard SVDD with the KDD CUP 1999 dataset. We used 300 normal instances randomly selected for training, and used all of data instances for testing. For the SVDD, the parameters σ and C were set to 0.4 and 0.05, respectively. For the proposed method, we used 80% of the eigenvalues and set σ to 100,000 in the K-PCA step of the training phase. The control parameter ε in (17) was set to 30 during the testing phase. The performance evaluation results are presented in **Table 1**.

The proposed method performed better with respect to the performance evaluation measurements, the exception being the detection rate of R2L relative to the standard SVDD. Note that it reveals a poor detection performance for R2L and U2R attacks compared with its detection rate for DoS and Probe attacks. The R2L and U2R attacks are host-based attacks that exploit the vulnerabilities of host computers, rather than network protocols. In R2L attacks, attackers connect to a remote host computer and attempt to exploit the vulnerability of a host computer to illicitly gain local access as a user. For U2R attacks, attackers access the system legally with a normal user account and attempt to obtain unauthorized local superuser privileges by exploiting vulnerabilities in the system. Hence, R2L and U2R are closely similar to the normal data in the KDD CUP 1999 dataset collected from network simulations [2, 25] . Once an attacker gains superuser privileges, every aspect of the system is under the attacker's control, and he or she can more easily gain access to related systems. Therefore, the U2R attacks are especially serious and dangerous. The proposed method led to a significant improvement in the detection rate of U2R attacks over the standard SVDD. The proposed approach also shows better or comparable performance compared to the others [10-11] .

Table 1. Experiment Results with the KDD CUP 1999 Dataset. The experimental results are yielded by the SVDD under the conditions σ=0.4, C=0.05, and the proposed method under the conditions σ=100000, ε=30. PRC_{Normal} is the precision of the normal dataset, and PRC_{Attack} is the precision of the attack dataset. DR_{DOS}, DR_{R2L}, DR_{U2R} and $DR_{\Pr ob}$ represent the attack-detection rate for DoS, R2L, U2R and Prob, respectively. F_{Normal} is the F-measure from the perspective of normal behavior, and F_{Attack} is the *F*-measure from the perspective of attack activity.

Measurements (%) \ **Algorithms**		**SVDD**	**ESVDD** [10]	**PESVDD** [11]	**The Proposed Approach**
Accuracy	*AC*	95.46	NA	NA	95.89
True Positive	*TPR*	95.22	NA	NA	95.71
False Positive	*FPR*	4.48	NA	NA	4.07
True Negative	*TNR*	95.52	NA	NA	95.93
False Negative	*FNR*	4.78	NA	NA	4.29
Precisions	PRC_{Normal}	83.71	NA	NA	85.06
	PRC_{Attack}	98.80	NA	NA	98.93
Attack	DR_{DOS}	**99.52**	**78**	**92**	**99.99**

Detection	DR_{R2L}	**38.01**	**80**	**84**	**37.75**
	DR_{U2R}	**88.64**	**89**	**95**	**96.59**
	DR_{Prob}	**100**	**67**	**78**	**100**
F-measure	F_{Normal}	89.10	NA	NA	90.07
	F_{Attack}	97.13	NA	NA	97.41

We empirically analyze the sensitivity of the parameter σ on the KDD CUP 1999 dataset. **Fig. 5** shows the changes of False Positive and False Negative value of the proposed methods with the parameter σ varying from 50000 to 200000. It should be noted that the selection of the parameter σ depends on a dataset. In the experiments of this paper, the value of the parameter σ was set empirically.

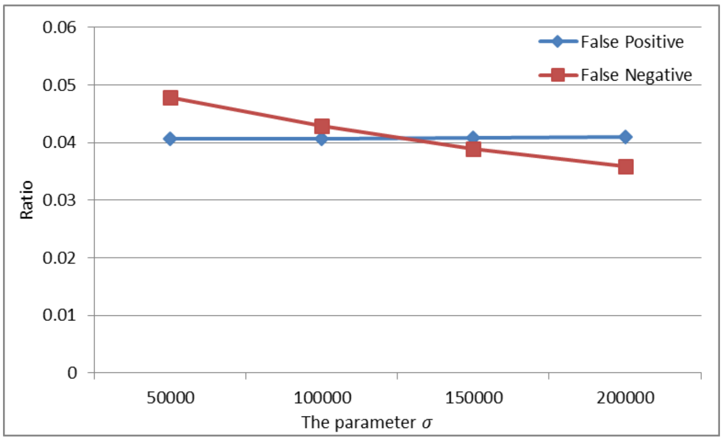


Fig. 5. Sensitivity analysis of the parameter σ on the KDD CUP 1999 dataset

4.4 SNMP MIB Traffic Flooding Dataset

To validate the proposed anomaly detection approach, we conducted experiments on a recently collected network traffic-flooding attack dataset. The SNMP MIB traffic-flooding dataset is a statistical dataset gathered from SNMP agents [3] . The dataset was collected from a testbed connected to a campus network. The normal dataset was collected from victim host computers and other host computers outside the testbed. The attack dataset was generated using the distributed denial of service attack tool [39] . The 13 MIB variables were selected with a correlation-based feature selection (CFS) method and gathered with an MIB update time prediction mechanism. The 13 MIB variables can be divided into 4 MIB groups: IP, TCP, UDP, and ICMP. The dataset was collected over 10 days of experimentation. The MIB variables of the target system were updated and gathered in an average of 15 seconds. The dataset consists of 5,000 MIB data instances, including 2,000 normal traffic instances, 1,000 TCP-SYN flooding attack instances, 1,000 UDP flooding attack instances, and 1,000 ICMP flooding attack instances. We used 300 normal instances randomly selected for training, and used all of the data instances for testing. For the SVDD, the parameters σ and C were set to 0.02 and 0.1, respectively. For the proposed method, we used 80% of the eigenvalues—the same as our experiment with the KDD CUP 1999 dataset—and set σ to 30 in the K-PCA training phase. The control parameter ε was set to 300 during the testing phase.

Table 2 provides the results for the performance evaluation on the traffic flooding attack. The proposed method outperformed the standard SVDD on the SNMP MIB traffic-flooding dataset. It achieved an overall accuracy rate of 99.88%, a true positive rate of 99.70%, and a true negative rate of 100%. The experimental results show that the proposed method is able not only to reduce the false-alarm rate, but also to improve the detection rate on the SNMP MIB traffic-flooding dataset.

Table 2. Experiment Results with the SNMP MIB Traffic Flooding Dataset. The experimental results are yielded by the SVDD under the conditions σ=0.02,C=0.1, and the proposed method under the conditions σ=30, ε=300. PRC_{Normal} is the precision of the normal dataset, and PRC_{Attack} is the precision of the attack dataset. $DR_{TCP-SYN}$, DR_{UDP} and DR_{ICMP} represent the attack-detection rate for the TCP-SYN flooding attack, UDP flooding attack, and ICMP flooding attack, respectively. F_{Normal} is the F-measure from the perspective of normal behavior, and F_{Attack} is the F-measure from the perspective of attack activity.

Measurements (%) \ **Algorithms**		**The SVDD**	**The Proposed Approach**
Accuracy	AC	99.56	99.88
True Positive	TPR	**99.55**	**99.70**
False Positive	FPR	0.43	0
True Negative	TNR	**99.57**	**100**
False Negative	FNR	0.45	0.3
Precisions	PRC_{Normal}	99.35	100
	PRC_{Attack}	99.70	99.90
Attack Detection	$DR_{TCP-SYN}$	99.40	100
	DR_{UDP}	99.30	100
	DR_{ICMP}	100	100
F-measure	F_{Normal}	99.45	99.85
	F_{Attack}	99.63	99.90

5. Conclusion

In this paper, we provided a new anomaly detection approach for anomaly intrusion detection based on a minimum volume enclosing ellipsoid and kernel principal component analysis to generate a better decision boundary around the normal behavior class of data. The proposed method mitigates the limitations in using the standard SVDD, i.e., the limited descriptive power of the hyper-sphere, by using the ellipsoid in feature space. Experimental results show the superiority of the proposed method. The efficacy of the proposed approach was demonstrated by depicting the decision boundaries of the standard SVDD with a Gaussian kernel function and the proposed method using a toy synthetic dataset. The proposed method creates a decision boundary that is more compact than it is with the standard SVDD. To validate the proposed method as an anomaly intrusion detection algorithm, we conducted experiments on the KDD CUP 1999 dataset and the SNMP MIB traffic flooding dataset. The proposed method led to a significant improvement in anomaly intrusion detection over the conventional SVDD approach.

References

[1] V. Chandola, A. Banerjee, V. Kumar, "Anomaly detection: A survey," *ACM Computing Survey*, vol.41, no.3, pp.15, 2009. Article (CrossRef Link).

[2] H. Lee, J. Song, D. Park, "Intrusion detection system based on multi-class SVM," in *Proc. of RSFDGrC, LNAI 3642*, pp.511-519, 2005. Article (CrossRef Link).

[3] J. Yu, H. Lee, M. Kim, D. Park, "Traffic flooding attack detection with SNMP MIB using SVM," *Computer Communications*, vol.31, no.17, pp.4212-4219, 2008. Article (CrossRef Link).

[4] S. Parsa and S.A. Naree, "A new semantic kernel function for online anomaly detection of software," *ETRI Journal*, vol.34, no.2, pp.288-291, 2012. Article (CrossRef Link).

[5] M. Kloft, P. Laskov, "Security analysis of online centroid anomaly detection," *JMLR*, vol.13, pp. 3681-3724, 2012.

[6] A. Banerjee, P. Burlina, R. Meth, "Fast hyperspectral anomaly detection via SVDD," in *Proc. of ICIP*, vol.4, pp.101-104, Sep. 16-19, 2007. Article (CrossRef Link).

[7] L. Jiaomin, W. Zhenzhou, F. Xinchun, W. Jing, "Intrusion detection technology based on SVDD," in *Proc. of ICINIS*, pp.15-18, 2009. Article (CrossRef Link).

[8] S.M. Guo, L.C. Chen, J.S.H. Tsai, "A boundary method for outlier detection based on support vector domain description," *Pattern Recognition*, vol.42, no.1, pp.77-83, 2009. Article (CrossRef Link).

[9] I. Kang, M.K. Jeong, D. Kong, "A differentiated one-class classification method with applications to intrusion detection," *Expert System with Applications*, vol.39, no.4, pp.3899-3905, 2012. Article (CrossRef Link).

[10] M. GhasemiGol, R. Monsefi, and H.S. Yazdi, "Intrusion detection by new data description method," in *Proc. of ISMS*, pp.1-5, 2010. Article (CrossRef Link).

[11] M. GhasemiGol, R. Monsefi, H.S. Yazdi, "Intrusion detection by ellipsoid boundary," *J. Netw. Syst. Manage*, vol.18, no.3, pp.265-282, 2010. Article (CrossRef Link).

[12] D.M.J. Tax, R.P.W. Duin, "Support vector data description," *Machine Learning*, vol.54, no.1, pp.45-66, 2004. Article (CrossRef Link).

[13] J. Park, J. Kim, H. Lee, D. Park, "One-class support vector learning and linear matrix inequalities," *IJFIS*, vol.3, no.1, 2003. Article (CrossRef Link).

[14] M. GhasemiGol, R. Monsefi, H.S. Yazdi, "Ellipse support vector data description," in *Proc. of EANN, CCIS 43*, pp.257–268, 2009. Article (CrossRef Link).

[15] D. Wang, D.S. Yeung, E.C.C. Tsang, "Structured one-class classification," *IEEE Trans. Syst., Man, Cybern. B*, vol.36, no.6, pp.1283–1295, 2006. Article (CrossRef Link).

[16] S. Rajasegarar, C. Leckie, J. C. Bezdek, M. Palaniswami, "Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks," *IEEE Trans. Inf. Forensics Security*, vol.5, no.3, pp.518-533, 2010. Article (CrossRef Link).

[17] P. G.-Teodoro, J. D.-Verdejo, G. M.-Fernandez, and E. Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenge," *Computers & Security*, vol.28, no.1-2, pp.18-28, 2009. Article (CrossRef Link).

[18] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, S. Zhou, "Specification-based anomaly detection: a new approach for detecting network intrusions," in *Proc. of ACM CCS*, pp.265-274, 2002. Article (CrossRef Link).

[19] C. Kruegel, F. Valeur, G. Vigna, R. Kemmerer, "Stateful intrusion detection for high-speed networks," in *Proc. of IEEE Symp. On Security and Privacy*, pp.285-293, 2002. Article (CrossRef Link).

[20] J.M. E.-Tapiador, P.G. Teodoro, J.E.D.-Verdejo, "Detection of web-based attacks through Markovian protocol parsing," in *Proc. of ISCC*, pp.457-462, 2005. Article (CrossRef Link).

[21] M. Ramadas, S. Ostermann, B. Tjaden, “Detecting anomalous network traffic with self-organizing maps,” in *Proc. of RAID, LNCS 2820*, pp.36-54, 2003. Article (CrossRef Link).

[22] Y. Yu and H. Wu, “Anomaly intrusion detection based upon data mining techniques and fuzzy logic,” in *Proc. of IEEE SMC*, pp.514-517, 2012. Article (CrossRef Link).

[23] M.S. Hoque, M.A. Mukit, and M.A.N. Bikas, “An implementation of intrusion detection system using genetic algorithm,” *IJNSA*, vol.4, no.2, pp.109-120, 2012. Article (CrossRef Link).

[24] Z. Mingqiang, H. Hui, W. Qian, “A graph-based clustering algorithm for anomaly intrusion detection,” in *Proc. of ICCSE*, pp.1311-1314, 2012. Article (CrossRef Link).

[25] H. Lee, Y. Chung, D. Park, “An adaptive intrusion detection algorithm based on clustering and kernel-method,” in *Proc. of PAKDD, LNAI 3918*, pp.603-610, 2006. Article (CrossRef Link).

[26] J.S.-Taylor, N. Cristianini, “*Kernel Methods for Pattern Analysis*,” pp.143-155, 2004.

[27] B. Schölkopf, A. Smola, K.-R. Müller, “Kernel principal component analysis,” in *Proc. of ICANNN, LNCS 1327*, pp.583-588, 1997.

[28] D.M.J. Tax, P. Juszczak, “Kernel whitening for one-class classification,” in *Pattern Recognition with Support Vector Machines, LNCS 2388*, pp.40-52, 2002. Article (CrossRef Link).

[29] S. Boyd and L. Vandenberghe, “*Convex optimization*, ” *Cambridge Univ*, 2004. Article (CrossRef Link).

[30] H. Hindi, “A tutorial on convex optimization”, in *Proc. of American Control Conference*, vol.4, pp.3252-3265, 2004. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.325.9447

[31] N. Moshtagh, “Minimum volume enclosing ellipsoids,” *GRASP Lab., Univ. of Pennsylvania*, 2005. Article (CrossRef Link).

[32] P. Sun, R.M. Freund, “Computation of minimum volume covering ellipsoids,” *Operations Research*, vol.52, no.5, pp.690-706, 2004. Article (CrossRef Link).

[33] P. Kumar, E.A. Yildirim, “Minimum volume enclosing ellipsoids and core set,” *Journal of Optimization Theory and Applications*, vol.126, no.1, pp.1-21, 2005. Article (CrossRef Link).

[34] L. Khachiyan, “Rounding of polytopes in the real number model of computation,” *Mathematics of Operations Research*, vol.21, no.2, pp.307-320, 1996. Article (CrossRef Link).

[35] UCI KDD Archive. KDD Cup 1999 Data. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[36] D.M.J. Tax, Data description toolbox (Dd_tools), 2013. Available: http://prlab.tudelft.nl/david-tax/dd_tools.html

[37] R.Kohavi, F. Provost, “Glossary of terms,” *Machine Learning*, vol.30, no.2/3, pp.271-274, 1998. Article (CrossRef Link).

[38] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, “A geometric framework for unsupervised anomaly detection,” in *Applications of Data Mining in Computer Security*, pp.77-101, 2002. Article (CrossRef Link).

[39] D. Dittrich, Distributed denial of service (DDoS) attacks/tools, Available: http://staff.washington.edu/dittrich/misc/ddos/

Hansung Lee received his BS, MS, and PhD degrees in computer science from Korea University, Sejong City, Rep. of Korea, in 1996, 2002, and 2008, respectively. From July 1996 to July 1999, he worked for DAEWOO engineering company, Seoul, Rep. of Korea. He was with Korea University, Sejong City, Rep. of Korea as a lecturer from March 2002 to February 2010. From November 2009 to Nobember 2014, he worked for Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea as a senior member of the research staff. He joined the Samsung Electronics Co., Ltd. Suwon, Rep. of Korea, in December 2014. His current research interests include pattern recognition, machine learning, optimization, data mining, and big-data analytics.

Daesung Moon received his MS degree in Dept. of Computer Engineering from Busan National University, Busan, Rep. of Korea, in 2001. He received his PhD degree in computer science from Korea University, Seoul, Rep. of Korea, in 2007. He joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea, in 2000, where he is currently a senior researcher. His research interests include network security, data mining, biometrics, and image processing.

Ikkyun Kim received the B.C.E., M.S.C.E. and Ph.D degrees in computer engineering from the Kyungpook National University, Daegue, Rep. of Korea, in 1994, 1996 and 2009 respectively. Dr. Kim has worked at Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea since 1996. He has developed several network-based intrusion detection systems and is currently working on new anomaly detection method against zero-day attacks for network security and Big-data analysis for security intelligence. His research interests include DDoS protection mechanism, high-speed network protection system, and cloud computing security.

Hoseok Jung received the PhD degree in computer science from Korea University, Sejong City, Rep. of Korea, in 2011. He is a chief in the Department of Knowledge & Information at the Korea Research Institute of Bioscience and Biotechnology (KRIBB), where he is working on information system, information security, and knowledge management. From 1992 to 1994, he was an application programmer in the Department of Medical Business, Korea Computer Cooperation, Seoul, Rep. of Korea. Since summer 1994, he has worked at the Korea Research Institute of Bioscience and Biotechnology, where he first served as application programmer and then as team manager. His research interests include data mining, machine learning, pattern recognition, and information retrieval.

Daihee Park received his BS degree in mathematics from Korea University, Seoul, Rep. of Korea, in 1982, and his PhD degree in computer science from Florida State University, USA, in 1992. He joined Korea University, Sejong City, Rep. of Korea in 1993, where he is currently a Professor in the Department of Computer and Information Science. His research interests include data mining and intelligent database.