2010 IEEE International Conference on Data Mining

Exploiting Local Data Uncertainty to Boost Global Outlier Detection

Bo Liu*, Jie Yin†, Yanshan Xiao*, Longbing Cao* and Philip S. Yu‡
*Faculty of Engineering and IT, QCIS
University of Technology, Sydney
Email: csbliu@gmail.com, syxiao@it.uts.edu.au, lbcao@it.uts.edu.au
†Information Engineering Laboratory
CSIRO ICT Centre, Australia
Email: Jie.Yin@csiro.au
‡Department of Computer Science
University of Illinois at Chicago
851 S. Morgan Street, Chicago, IL 60607-0753
Email: psyu@cs.uic.edu

***Abstract*—This paper presents a novel hybrid approach to outlier detection by incorporating local data uncertainty into the construction of a global classifier. To deal with local data uncertainty, we introduce a confidence value to each data example in the training data, which measures the strength of the corresponding class label. Our proposed method works in two steps. Firstly, we generate a pseudo training dataset by computing a confidence value of each input example on its class label. We present two different mechanisms: kernel k-means clustering algorithm and kernel LOF-based algorithm, to compute the confidence values based on the local data behavior. Secondly, we construct a global classifier for outlier detection by generalizing the SVDD-based learning framework to incorporate both positive and negative examples as well as their associated confidence values. By integrating local and global outlier detection, our proposed method explicitly handles the uncertainty of the input data and enhances the ability of SVDD in reducing the sensitivity to noise. Extensive experiments on real life datasets demonstrate that our proposed method can achieve a better tradeoff between detection rate and false alarm rate as compared to four state-of-the-art outlier detection algorithms.**

***Keywords*-Outlier detection; Data uncertainty; SVDD**

I. Introduction

Outlier detection has attracted increasing attention in machine learning and data mining areas due to its wide-ranging applications from machine fault detection, credit card fraud detection, network intrusion to medical diagnosis. Outliers refer to the data objects that are markedly different from or inconsistent with the remaining set of data [8], [13]. Traditional outlier detection algorithms typically assume that outliers are difficult or costly to obtain due to their rare occurrences in real-world applications. Therefore, most of previous approaches mainly focus on modelling a representation of the normal data so as to identify outliers that do not fit the model well.

Depending on the nature of representation models, previous approaches to outlier detection can be classified into four broad categories: (1) distribution-based approaches [4], in which a pre-specified probability distribution is usually used to model the normal data and then a statistical test is applied to detect if a data point is an outlier; (2) density-based approaches [3], [6], [12], in which local outliers are identified by examining the distances to their nearest neighbors; (3) clustering-based approaches [14], which find outliers as by-product of a clustering algorithm; (4) model-based approaches [18], which typically use a predictive model to characterize the normal data and detect outliers as deviations from the model. In this category, the support vector data description (SVDD) proposed by Tax and Duin [25], [26] has been demonstrated to be capable of detecting outliers in various application domains.

Despite much progress in this area, most of the existing works on outlier detection have not explicitly dealt with the uncertainty of the input data. An underlying assumption is that the training dataset is perfectly labeled for building outlier detection models or classifiers. However, in many real-world applications, the data may be corrupted with noise or may only be partially complete [2], [5]. For example, sensor networks typically generate a large amount of uncertain data subject to sampling errors or instrument imperfections. Thus, a normal example may behave like an outlier, even though the example itself may not be an outlier. Such uncertain information might introduce labeling imperfections or errors into the training data, which further limits the accuracy of subsequent outlier detection. Moreover, another important observation is that, negative examples or outliers, although very few, do exist in many applications. For example, in the network intrusion domain, in addition to extensive data about the normal traffic conditions in the network, there also exist a small number of cyber attacks that can be collected to facilitate outlier detection. Although these outliers are not sufficient for constructing a binary classifier, they can be incorporated into the training process to refine the decision boundary around the normal data for outlier detection.

In this paper, we address the problem of outlier detection

1550-4786/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDM.2010.10

with very few labeled negative examples. In order to cope with data uncertainty, we propose a novel hybrid approach to outlier detection by generalizing the SVDD learning framework on imperfectly labeled training dataset. Specifically, we associate each example in the training dataset not only with a class label but also a confidence value which measures the strength of the corresponding label. Our proposed approach works in two steps. In the first step, we generate a pseudo training dataset by computing a confidence value of each input example on its class label based on the local data behavior. Two different mechanisms are proposed to generate the confidence values: kernel k-means clustering methods and kernel LOF-based method. In the second step, we construct a global classifier for outlier detection by generalizing the SVDD-based learning process. Associated with a confidence value, each data point can have different contributions to the learning of the decision boundary. By integrating local and global outlier detection, our proposed method explicitly handles the uncertainty of the input data and enables the ability of SVDD in reducing the sensitivity to noise. Extensive experiments on real life datasets show that our proposed method can offer a better tradeoff between detection rate and false alarm rate as compared to three state-of-the-art outlier detection algorithms.

The rest of the paper is organized as follows. Section II discusses previous works related to our outlier detection problem. Section III presents our proposed method to outlier detection in detail. Section IV reports extensive experimental results on real-world datasets. Section V concludes the paper and discusses possible directions for future work.

II. Related Work

In this section, we discuss previous work related to our outlier detection problem in three parts. In Section II-A, we first review previous work on outlier detection in the data mining area. In Section II-B, we discuss another branch of related work on learning from imbalanced data and cost-sensitive learning. Finally, in Section II-C, we give a brief description of support vector data description.

A. Outlier Detection

Outlier detection techniques can be classified into four categories: distribution-based approaches [10], density-based approaches [6], [12], clustering-based approaches [14], [22] and model-based approaches [8]. Distribution-based approaches [10] are the earliest algorithms developed for outlier detection, which fit a statistical model (e.g. Normal, Poisson, Gaussian, etc.) to the normal data and then apply a statistical test to determine if an unseen data point belongs to this model or not. Points that have low probability of belonging to the learned model are detected as outliers. The key disadvantage of distribution-based approaches is that they rely on the assumption that the data is generated from a particular distribution. However, this assumption often does not hold true in practice, especially for high dimensional real data sets.

For density-based approaches [6], [12], the main task is to define pairwise distances between data points and identify outliers by examining the distance or relative density of each data point to its local neighbors. Representative methods include LOF (Local Outlier Factor) [6] and its variants, which assign an outlier score to any given data point, depending on its distances in the local neighborhood. The advantage of these approaches is that they do not make any assumption for the generative distribution of the data. However, these approaches incur a high computational complexity in the testing phase, since they involve calculating the distance between each test instance and all the other instances to compute nearest neighbors.

Clustering-based approaches [14], [22] mainly rely on applying clustering techniques to characterize the local data behavior. As a by-product of clustering, small clusters that contain significantly less data points than other clusters are considered as outliers. Clustering-based approaches are unsupervised in nature without requiring any labeled training data. However, the performance of unsupervised outlier detection is limited.

Model-based approaches [15] typically characterize the normal data via a predictive model and detect outliers as deviations from the learned model. Among others, support vector data description (SVDD) [25], [26] has been demonstrated empirically to be capable of detecting outliers in various domains. The most attractive feature of SVDD is that it can transform the original data into a feature space and detect global outliers more effectively for high-dimensional data. However, its performance is sensitive to the noise involved in the input data.

Depending on the availability of a training dataset, outlier detection techniques described above operate in two different modes: supervised and unsupervised. Distribution-based approaches and model-based approaches fall into the category of supervised outlier detection, which assumes the availability of a training dataset that has labeled instances for normal class (as well as anomaly class sometimes). For supervised outlier detection, obtaining accurate and representative labels for the training dataset, especially for the anomaly class is usually very challenging. Several techniques [1], [23], [27] have been proposed that inject artificial anomalies into a normal dataset to obtain a labeled training data set. In addition, the work of [24] presents a new method to detect outliers by utilizing the instability of the output of a classifier built on bootstrapped training data.

The method we propose in this work is a hybrid approach to outlier detection, which captures local data uncertainty by generating the confidence of each input example on its class label based on the local neighborhood. Such information is then incorporated into the generalized SVDD framework to enhance a global classifier for outlier detection.

B. Difference from Imbalanced Data Classification

The outlier detection problem that we consider in this paper is also related to the problem of imbalanced data classification [9], in which outliers corresponding to the negative class are extremely small in proportion as compared to the normal data corresponding to the positive class.

Research on imbalanced data classification falls into two main categories. The first category attempts to modify the class distribution of training data before applying any learning algorithms [7]. This is usually done by over-sampling, which replicates the data in the minority class, or under-sampling, which throws away part of the data in the majority class. The second category focuses on making a particular classifier learner cost sensitive, by setting the false positive and false negative costs very differently and incorporating the cost factors into the learning process [9]. Representative methods include cost-sensitive decision trees [16] and cost-sensitive SVMs [11], [19]. When imbalanced data are present, researchers have argued for the use of ranking-based metrics, such as the ROC curve and the area under ROC curve (AUC) [20] instead of using accuracy.

The difference between imbalanced data classification and our outlier detection problem is that: in imbalanced data classification, the examples from one or more minority classes are often self-similar, potentially forming compact clusters, while in outlier detection, the outliers are typically scattered so that the distribution of the negative class cannot be well represented by the very few negative training examples. To solve our problem, we can exploit cost-sensitive learning algorithms, but the false positive and false negative costs are usually unknown to us in real life applications. Therefore, we exploit a novel one-class classification method for outlier detection, which aims at building a decision boundary around the normal data, and utilize the few negative examples to refine the boundary.

C. Support Vector Data Description

The Support Vector Data Description (SVDD) [25] is one of the best-known support vector learning methods for one-class classification. Given a set of target data $\{\mathbf{x}_i\}, i = 1, \ldots, l$, where $\mathbf{x}_i \in R^m$, the basic idea is to find a sphere that contains most of target data such that its corresponding radius R is minimized:

$$\begin{aligned} \min \quad & F(R, \mathbf{o}, \xi_i) = R^2 + C \sum_{i=1}^{l} \xi_i, \\ s.t. \quad & \| \mathbf{x}_i - \mathbf{o} \|^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, \end{aligned} \tag{1}$$

where slack variables ξ_i are introduced to allow some data points to lie outside the sphere, and $C > 0$ controls the tradeoff between the volume of the sphere and the number of errors. $\sum_{i=1}^{l} \xi_i$ means the penalty for misclassified patterns.

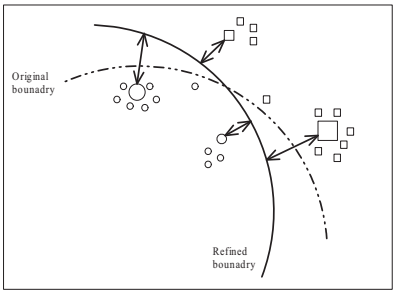


Figure 1. Motivation behind our proposed approach

By introducing Lagrange multipliers, the above optimization problem is transformed into the dual formulation:

$$\begin{aligned} \max \quad & \sum_{i=1}^{l} \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i=1}^{l} \sum_{k=1}^{l} \alpha_i \alpha_k (\mathbf{x}_i \cdot \mathbf{x}_k) \\ s.t. \quad & 0 \leq \alpha_i \leq C, \ \sum_i \alpha_i = 1. \end{aligned} \tag{2}$$

The solution of Equation (2) gives a set of $\{\alpha_i\}$. Data points with $\alpha_i > 0$ are called the support vectors of the description. For a test point $\mathbf{x}$, the distance to the center of the sphere is calculated as: $\| \mathbf{x} - \mathbf{o} \|^2 = (\mathbf{x} \cdot \mathbf{x}) - 2 \sum_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)$. The point $\mathbf{x}$ is classified as normal data when this distance is less than or equal to the radius R. Otherwise, it is flagged as an outlier.

To allow a more flexible description, the original data points are typically mapped into a feature space via a nonlinear mapping function $\phi(\cdot)$. The mapping is performed implicitly by replacing the inner products $(\cdot, \cdot)$ in Equation (2) by a kernel function $K(\mathbf{x}, \mathbf{x}_i) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i)$. The most attractive feature of SVDD is that it can transform the input data into a feature space and detect global outliers effectively for high-dimensional data. However, its performance is sensitive to the noise involved in the input data. Our proposed method generalizes SVDD to incorporate the confidence of class labels into the training process, which mitigates the effect of noise on outlier detection.

III. Our Proposed Algorithm

In this section, we provide a detailed description about our proposed approach to outlier detection. Given a set of training data D which consists of l normal examples and a small amount of n outlier (or abnormal) examples, the objective is to build a classifier using both normal and abnormal training data and the classifier is thereafter applied to classify unseen test data. However, subject to sampling errors or device imperfections, an normal example may behave like an outlier, even though the example itself may not be an outlier. Such error factors might result in an imperfectly labeled training data, based on which the subsequent outlier detection becomes grossly inaccurate.

To deal with label imperfections, we propose to associate each input data with a confidence value, which indicates the likelihood of an input data belonging to its corresponding class label. Such information is thereafter incorporated into the construction of a global classifier for outlier detection. The motivation behind our proposed method is illustrated in Figure 1. In the figure, positive examples are depicted as circles and negative examples squares. The size of the circles/squares indicates their associated confidence values. Intuitively, the higher confidence we have on a label, the larger force we want to have on that sample towards the decision boundary. The dashed line is the original decision boundary derived from the standard SVDD training, and the solid line is the refined decision boundary after taking labels' confidence values into consideration.

Based on this idea, our proposed method works in two steps as follows:

- In the first step, we generate a *pseudo training dataset* by computing a confidence value for each input data on its class label based on local data behavior.
- In the second step, we construct a global SVDD-based classifier for outlier detection by using both normal and abnormal examples as well as the confidence values associated with their class labels.

In the following, we describe the two steps in detail.

A. Computing Confidence on the Class Labels

The main task of this step is to create a pseudo training dataset by computing a confidence value for each input data on its class label. The generated pseudo training data consists of two parts: $(\mathbf{x}_1, m^T(\mathbf{x}_1)), \ldots, (\mathbf{x}_l, m^T(\mathbf{x}_l))$ for l normal examples and $(\mathbf{x}_{l+1}, m^N(\mathbf{x}_{l+1})), \ldots, (\mathbf{x}_{l+n}, m^N(\mathbf{x}_{l+n}))$ for n abnormal examples, were $m^T(\mathbf{x}_1)$ and $m^N(\mathbf{x}_j)$ indicate the likelihood of example $\mathbf{x}$ belonging to the the normal class and the outlier class, respectively.

We propose two different schemes to compute a confidence value for each input data, inspired by clustering-based and density-based approaches to outlier detection. The basic idea is to capture the local data uncertainty by examining the relative distances of each input data to its local neighbors.

1) Kernel K-Means Clustering Method: We adopt the kernel k-means clustering algorithm to generate a confidence value for each input data. Based on a nonlinear mapping function $\phi : \mathbf{x} \to \phi(\mathbf{x})$, kernel k-means clustering minimizes the following objective function:

$$J = \sum_{i=1}^{k} \sum_{j=1}^{l+n} \|\phi(\mathbf{x}_j) - \phi(\mathbf{v}_i)\|^2, \tag{3}$$

where k is the number of clusters and $\mathbf{v}_i$ is the cluster center of the i^{th} cluster.

By solving this optimization problem, k-means clustering returns a set of local clusters, in which data points belonging to a same cluster are more similar to each other. Intuitively, for a data point, if most of data point in the same cluster are normal, it would have a high probability of being normal, and if there is an outlying point that doe not belong to any cluster, it would have a high probability of being an outlier. Therefore, we calculate the confidence values as follows. For a given cluster j, assume there exist l_j^p normal examples and l_j^n negative examples. The confidence value of a normal example belonging to the normal class is calculated as $m^T(\mathbf{x}_t) = l_j^p / l_j^p + l_j^n$. Similarly, the confidence value of an abnormal example belonging to the negative class is computed as $m^N(\mathbf{x}_n) = l_j^n / l_j^p + l_j^n$.

The advantage of kernel k-means is that it can partition the dataset into a set of local clusters that are non-linearly separable in the input space. However, the main limitation is that it does not work well on datasets with varying densities by using a global distance function, which causes the generated confidence values to be inaccurate.

2) Kernel LOF-based Method: To cope with datasets with varying densities, we propose a local density-based method to compute a confidence value for each input data. Inspired by the LOF algorithm [6], the basic idea is to examine the relative distance of a point to its local neighbors. Specifically, we extend the original LOF into the kernel space by using kernel methods and generate the confidence values in the kernel space instead of the input space.

For each point $\mathbf{x}_i$, we first compute its local reachability density, which is the average reachability distance based on the k-nearest neighbors of $\mathbf{x}_i$.

$$lrd_k(\mathbf{x}_i) = \frac{1}{k} \sum_{\mathbf{x}_j \in N_k(\mathbf{x}_i)} \text{reach-list}_k(\mathbf{x}_i, \mathbf{x}_j), \tag{4}$$

where $N_k(\mathbf{x}_i)$ is a set of k-nearest neighbors of point $\mathbf{x}_i$. Here, reach-list$_k(\mathbf{x}_i, \mathbf{x}_j)$ denotes the reachability distance of object $\mathbf{x}_i$ with respect to object $\mathbf{x}_j$ in the feature space. It is computed as the larger value between A and B, where A is the actual distance between $\mathbf{x}_j$ and $\mathbf{x}_i$, and B is the distance between $\mathbf{x}_i$ and its kth nearest neighbor. Interested readers please refer to [6] for detailed definitions.

After the local reachability density $lrd_k(\mathbf{x}_i)$ is computed, for the point $\mathbf{x}_i$, we find its lrd-neighborhood $N_{lrd}(\mathbf{x}_i) = \{\mathbf{x}_j \in D \mid \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 \leq lrd_k(\mathbf{x}_i)\}$. The distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the feature space is computed as

$$\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 = K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j). \tag{5}$$

For a positive sample, suppose that there exist l_t examples out of $|N_{lrd}(\mathbf{x}_i)|$ nearest neighbors belonging to the positive class. The confidence value of $\mathbf{x}_t$ towards positive class is defined as $m^T(\mathbf{x}_t) = l_t / |N_{lrd}(\mathbf{x}_i)|$, where $|N_{lrd}(\mathbf{x}_i)|$ denotes the number of nearest neighbors in the lrd-neighborhood. Similarly, for a negative example, assume there exist l_n examples out of $|N_{lrd}(\mathbf{x}_i)|$ nearest neighbors belonging to the negative class, The confidence value of $\mathbf{x}_n$ towards negative class in feature space is given as $m^N(\mathbf{x}_n) = l_n / |N_{lrd}(\mathbf{x}_i)|$.

B. Constructing Soft-SVDD Classifiers

After generating a pseudo training dataset, the next step is to build a global SVDD-based classifier for outlier detection. Below, we give a new formulation of SVDD by using both normal and abnormal data as well as the associated confidence on the class labels.

1) Primal Formulation: Since the membership functions $m^T(\mathbf{x}_i)$ and $m^N(\mathbf{x}_j)$ indicate the degree of the belongingness of data example $\mathbf{x}_i$ toward target class and negative class, the solution to soft-SVDD can be achieved by solving the following optimization problem:

$$\begin{aligned} \min \quad & F = R^2 + C_1 \sum_{i=1}^{l} m^T(\mathbf{x}_i)\xi_i + C_2 \sum_{j=l+1}^{l+n} m^N(\mathbf{x}_j)\xi_j \\ s.t. \quad & \| \mathbf{x}_i - \mathbf{o} \|^2 \leq R^2 + \xi_i, \\ & \| \mathbf{x}_j - \mathbf{o} \|^2 \geq R^2 - \xi_j, \\ & \xi_i \geq 0,\ \xi_j \geq 0, \end{aligned} \tag{6}$$

Above, Parameters C_1 and C_2 control the tradeoff between the sphere volume and the errors. Parameters ξ_i are ξ_j are defined as a measure of error, as in SVDD. The terms $m^T(\mathbf{x}_i)\xi_i$ and $m^N(\mathbf{x}_j)\xi_j$ can be therefore considered as a measure of error with different weighing factors. Note that a smaller value of $m^T(\mathbf{x}_i)$ could reduce the effect of the parameter ξ_i in Equation (6), such that the corresponding data example $\mathbf{x}_i$ becomes less significant in the training.

2) Dual Problem: To solve the above optimization problem, we introduce Lagrange multipliers $\alpha_i^T \geq 0$, $\alpha_j^N \geq 0$, $\beta_i^T \geq 0$, $\beta_j^N \geq 0$, and convert problem (6) into problem (7).

$$\begin{aligned} L \quad = \quad & R^2 + C_1 \sum_{i=1}^{l} m^T(\mathbf{x}_i)\xi_i + C_2 \sum_{j=l+1}^{l+n} m^N(\mathbf{x}_j)\xi_j \\ - \quad & \sum_{i=1}^{l} \alpha_i^T (R^2 + \xi_i - \| \Phi(\mathbf{x}_i) - \mathbf{o} \|^2) - \sum_{j=l+1}^{l+n} \beta_j^N \xi_j \\ - \quad & \sum_{i=1}^{l} \beta_i^T \xi_i - \sum_{j=l+1}^{l+n} \alpha_j^N (\| \Phi(\mathbf{x}_j) - \mathbf{o} \|^2 - R^2 - \xi_j). \end{aligned} \tag{7}$$

Setting the partial derivatives of L with respect to $R, \mathbf{o}, \xi_i, \xi_j$ equal to zeros respectively, we can obtain

$$\begin{aligned} \frac{\partial L}{\partial R} = 0 \quad & \longrightarrow \quad \sum_{i=1}^{l} \alpha_i^T - \sum_{j=l+1}^{l+n} \alpha_j^N = 1, \\ \frac{\partial L}{\partial \mathbf{o}} = 0 \quad & \longrightarrow \quad \sum_{i=1}^{l} \alpha_i^T (\mathbf{o} - \Phi(\mathbf{x}_i)) = \sum_{j=l+1}^{l+n} \alpha_j^N (\mathbf{o} - \Phi(\mathbf{x}_j)), \\ \frac{\partial L}{\partial \xi_i} = 0 \quad & \longrightarrow \quad \alpha_i^T + \beta_i^T = C_1 m^T(\mathbf{x}_i), \\ \frac{\partial L}{\partial \xi_j} = 0 \quad & \longrightarrow \quad \alpha_j^N + \beta_j^N = C_2 m^N(\mathbf{x}_j). \end{aligned}$$

Replacing these into Equation (7), we get the following dual formulation:

$$\begin{aligned} \max \quad & \sum_{i=1}^{l} \alpha_i^T K(\mathbf{x}_i, \mathbf{x}_i) + 2 \sum_{i=1}^{l} \sum_{j=l+1}^{l+n} \alpha_i^T \alpha_j^N K(\mathbf{x}_i, \mathbf{x}_j) \\ & - \sum_{j=l+1}^{l+n} \alpha_j^N K(\mathbf{x}_j, \mathbf{x}_j) - \sum_{i=1}^{l} \sum_{k=1}^{l} \alpha_i^T \alpha_k^T K(\mathbf{x}_i, \mathbf{x}_k) \\ & - \sum_{j=l+1}^{l+n} \sum_{v=l+1}^{l+n} \alpha_j^N \alpha_v^N K(\mathbf{x}_j, \mathbf{x}_v) \\ s.t. \quad & 0 \leq \alpha_i^T \leq C_1 m^T(\mathbf{x}_i), \\ & 0 \leq \alpha_j^N \leq C_2 m^N(\mathbf{x}_j), \\ & \sum_{i=1}^{l} \alpha_i^T - \sum_{j=l+1}^{l+n} \alpha_j^N = 1. \end{aligned} \tag{8}$$

By setting $\alpha_i = \alpha_i^T (i = 1, 2, \ldots, l)$, $\alpha_i = \alpha_i^N (i = l + 1, l + 2, \ldots, l + n)$, $C_i = C_1 m^T(\mathbf{x}_i) (i = 1, 2, \ldots, l)$ and $C_i = C_2 m^N(\mathbf{x}_i) (i = l+1, l+2, \ldots, l+n)$, the optimization Problem (8) is rewritten as follows:

$$\begin{aligned} \max \quad & \sum_{i=1}^{l+n} \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i=1}^{l+n} \sum_{j=1}^{l+n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ s.t. \quad & 0 \leq \alpha_i \leq C_i \quad i = 1, 2, \ldots, l+n, \\ & \sum_{i=1}^{l+n} \alpha_i = 1. \end{aligned} \tag{9}$$

After solving the above dual problem, we obtain the Lagrange multipliers $\alpha_i (1 \leq i \leq l + n)$, which give the centroid of the minimum sphere as a linear combination of $\mathbf{x}_i$:

$$\mathbf{o} = \sum_{i=1}^{l+n} \alpha_i \Phi(\mathbf{x}_i). \tag{10}$$

Above, we find only the patterns with $\alpha_i \neq 0$ construct the centroid of the minimum sphere, and these pattern are called support vectors.

3) Decision Boundary Construction: By applying Karush-Kuhn-Tucker conditions [28], we then obtain the radius R of the decision hyperplane. Assume $\mathbf{x}_u$ is one of the patterns lying on the surface of sphere, R can be calculated as follows:

$$\begin{aligned} R^2 \quad = \quad & \|\mathbf{x}_u - \mathbf{o}\|^2 \\ = \quad & K(\mathbf{x}_u, \mathbf{x}_u) + (\mathbf{o}, \mathbf{o}) - 2K(\mathbf{x}_u, \mathbf{o}) \\ = \quad & K(\mathbf{x}_u, \mathbf{x}_u) + \sum_{i=1}^{l+n} \sum_{k=1}^{l+n} \alpha_i \alpha_k K(\mathbf{x}_i, \mathbf{x}_k) \\ & -2 \sum_{i=1}^{l+n} \alpha_i (\mathbf{x}_i, \mathbf{x}_u) \end{aligned} \tag{11}$$

To classify a test point $\mathbf{x}$, we just calculate its distance to the centroid of the hypersphere. If this distance is less than or equal to R, i.e.

$$\|\mathbf{x} - \mathbf{o}\|^2 \leq R^2, \tag{12}$$

$\mathbf{x}$ is accepted as the normal data. Otherwise, it is detected as an outlier.

4) Complexity Analysis: The computational complexity of solving the optimization Problem (9) is $O(l+n)^2$. Since outliers only take a very small portion of the training set, i.e. $n \ll l$, Soft-SVDD has approximately the same complexity as the standard SVDD ($O(l^2)$).

IV. Experimental Evaluation

To validate the effectiveness of our proposed method, we perform extensive experiments on 10 real life datasets. For all reported results, the test platform is a Dual 2.8GHz Intel Core2 T9600 PC with 3.45GB RAM.

A. Baselines and Metrics

We implemented two variants of our proposed method using two mechanisms to compute the confidence values: kernel k-means clustering and kernel LOF, which are referred to as CLU-Soft-SVDD and LOF-Soft-SVDD, respectively. For comparison, four state-of-the-art outlier detection algorithms are used as baselines.

1) The first one is kernel k-means clustering [22], [8] which finds outliers from resulting clusters in the feature space.
2) The second one is the kernel-LOF algorithm, which generalizes the LOF algorithm [6] by computing the outlier factor in the feature space.

The first two baselines are used to show the improvement of our proposed method over clustering-based and density-based approaches to outlier detection.

3) The third one is SVDD [25], which builds a one-class classifier solely based on the normal data. This baseline is used to test the ability of our proposed method in reducing the sensitivity of SVDD to noise.
4) The fourth algorithm is the cost-sensitive SVM (CS-SVM) [19], which assigns different costs to the normal data and abnormal data so as to learn a binary classifier for outlier detection. This baseline is used to test the effectiveness of our proposed method when very few labeled negative examples are available for training.

The performance of outlier detection algorithms can be evaluated based on two error rates: *detection rate* and *false alarm rate*. The detection rate is computed as the ratio of the number of correctly detected outliers to the total number of outliers. The false alarm rate is computed as the ratio of the number of normal examples that are incorrectly detected as outliers to the total number of normal examples. We compare the six algorithms using the ROC curve which plots the detection rate against the false alarm rate. We also explicitly compute the AUC values [20] to compare the six algorithms. A desirable algorithm with a high detection rate and a low false alarm rate should have an AUC value closer to one.

B. Datasets and Parameter Settings

In our experiments, we used 10 real life datasets that have been used earlier by other researchers for outlier detection [17], [29]. These datasets include Abalone class 1-8, Spambase other, Thyriod hyperfunction, Waveform 1, Satellite Grey soil, Delft pump 5x1, Diabetes present, Breast Wisconsin, Heart Cleveland, and Arrhythmia normal [1]. To perform outlier detection with very few abnormal data, we randomly selected 50% of positive data and a small number of abnormal data for training, such that 95% percent of the training data belong to the positive class and only 5% percent belong to the negative class. All the remaining data are used for testing.

For all the algorithms, the Gaussian RBF kernel was used in the experiments

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\| \mathbf{x} - \mathbf{x}_i \|^2 / 2\sigma^2). \tag{13}$$

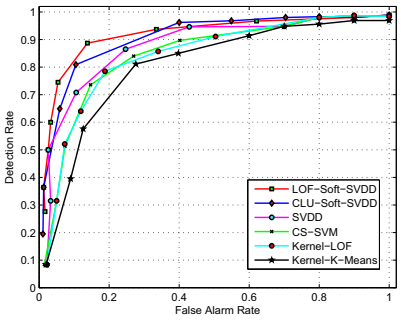
We used cross-validation on the training data to tune the parameters for CLU-Soft-SVDD, LOF-Soft-SVDD, SVDD and CS-SVM. The parameter σ in the RBF kernel was searched in the range from 2^{-3} to 2^4. In addition, the parameter C in SVDD, as well as C_1, C_2 in Soft-SVDD and CS-SVM was selected from 2^0 to 2^4. All the reported ROC and AUC results are based on this setting.

For CLU-Soft-SVDD and kernel k-means, we varied the number of clusters from 2 to $\frac{l+n}{2}$ and obtained the optimal number of clusters k^* by minimizing the external criteria in [21]. For LOF-Soft-SVDD, we set the number of nearest neighbors k used for computing confidence values to the number of negative samples in the training set. For kernel LOF, we followed the experimental setting in [6] to compute the maximum LOF by varying k in the range from 30 to 50.

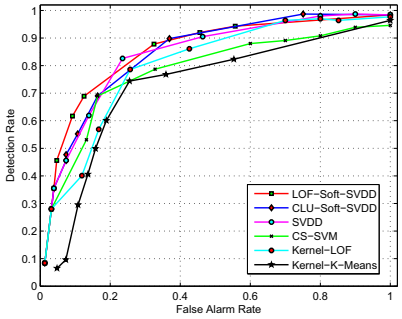
C. Classification Accuracy

We first performed experiments to compare the classification accuracy of the six algorithms. For each dataset, we generated the training data by randomly selecting positive examples and negative examples at the ratio of 95% to 5%, and applied the six algorithms to the training data and evaluated the performance on the remaining test data. Figure 2 shows the ROC curves for six out of 10 datasets. Our proposed method, CLU-Soft-SVDD and LOF-Soft-SVDD, can be observed to outperform other baselines. Note that, due to limited space, we only show the ROC curves for six datasets, and in the following experiments, we will also report detailed analysis results for six datasets. However, the reported results are all consistent on the 10 datasets.

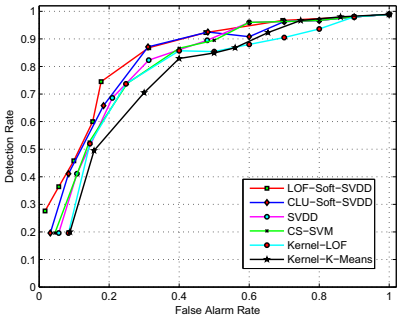
[1]The 10 datasets used in our experiments are all available online from http://homepage.tudelft.nl/n9d04/occ/index.html



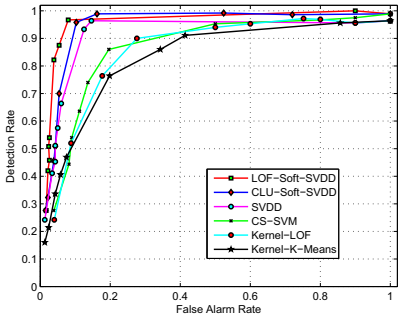
(a) Abalone



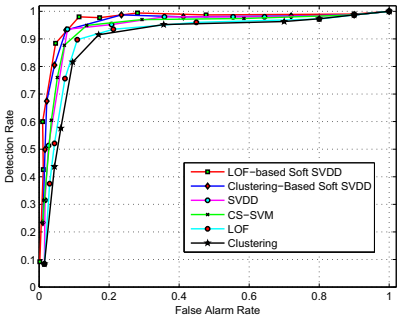
(b) Spambase



(c) Thyroid



(d) Waveform



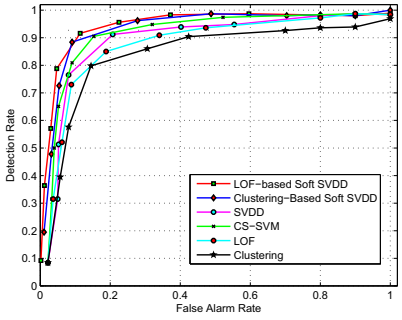


Table I
AVERAGE AUC VALUES AND THE STANDARD DEVIATIONS ON THE 10 DATASETS

Datasets	CLU-Soft-SVDD	LOF-Soft-SVDD	SVDD	CS-SVM	Kernel-LOF	Kernel k-Means
Abalone	0.894±0.033	**0.911±0.035**	0.878±0.041	0.855±0.044	0.848±0.041	0.842±0.048
Spambase	0.849±0.059	**0.866±0.054**	0.830±0.076	0.814±0.079	0.808±0.082	0.749±0.090
Thyroid	0.821±0.045	**0.840±0.042**	0.806±0.042	0.812±0.050	0.769±0.058	0.756±0.068
Waveform	0.924±0.036	**0.936±0.026**	0.919±0.045	0.866±0.049	0.837±0.069	0.815±0.069
Satellite Grey soil	0.935±0.044	**0.944±0.045**	0.907±0.052	0.917±0.050	0.870±0.058	0.859±0.062
Delf Pump	0.961±0.057	**0.963±0.039**	0.948±0.042	0.942±0.060	0.938±0.072	0.927±0.079
Diabetes	0.753±0.072	**0.769±0.068**	0.736±0.078	0.726±0.089	0.652±0.098	0.601±0.116
B.Wisconsin	0.958±0.033	**0.977±0.038**	0.943±0.039	0.942±0.043	0.908±0.050	0.873±0.078
H.Cleveland	0.753±0.059	**0.797±0.078**	0.728±0.098	0.746±0.073	0.674±0.127	0.648±0.128
Arrhythmia	0.903±0.063	**0.913±0.052**	0.853±0.073	0.872±0.063	0.832±0.081	0.798±0.079

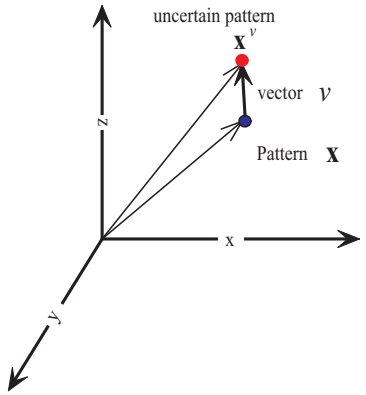


Figure 3. Illustration of the method used to add the noise to a data example: $\mathbf{x}$ is an original data example, $\mathbf{v}$ is a noise vector, $\mathbf{x^v}$ is the new data example with added noise. Here we have $\mathbf{x^v} = \mathbf{x} + \mathbf{v}$.

Figure 3 illustrates the basic idea of the method used to add the noise to data examples. Specifically, the standard deviation σ_i^0 of the entire data along the ith dimension was first obtained. In order to model the difference in noise on different dimensions, we defined the standard deviation σ_i along the ith dimension, whose value was randomly drawn from the range $[0, 2 \cdot \sigma_i^0]$. Then, for the ith dimension, we added noise from a random distribution with standard deviation σ_i. In this way, a data example $\mathbf{x}_j$ was added with the noise, which can be presented as a vector

$$\sigma^{\mathbf{x}_j} = [\sigma_1^{\mathbf{x}_j}, \sigma_2^{\mathbf{x}_j}, \cdots, \sigma_{n-1}^{\mathbf{x}_j}, \sigma_n^{\mathbf{x}_j}]. \quad (14)$$

Here, n denotes the number of dimensions for a data example $\mathbf{x}_j$, and $\sigma_i^{\mathbf{x}_j}$, $i = 1, \cdots n$ represents the noise added into the ith dimension of the data example.

In our experiments, we made the percentage of the data corrupted by noise vary from 0% to 30%, and applied the six methods on these datasets. Figure 4 shows the AUC values achieved by the six algorithms with respect to different percentages of training data corrupted by noise. We can see that, as more noise is added into the training data, the overall performance of the six methods degrades. This occurs because, when more noise is involved, target class becomes more indistinguishable from negative class. However, we can clearly see that, the two methods, LOF-soft-SVDD and CLU- soft-SVDD, can still consistently yield higher accuracy than kernel LOF, kernel k-means, SVDD, and CS-SVM. This concludes that, our proposed soft-SVDD can effectively reduce the effect of noise.

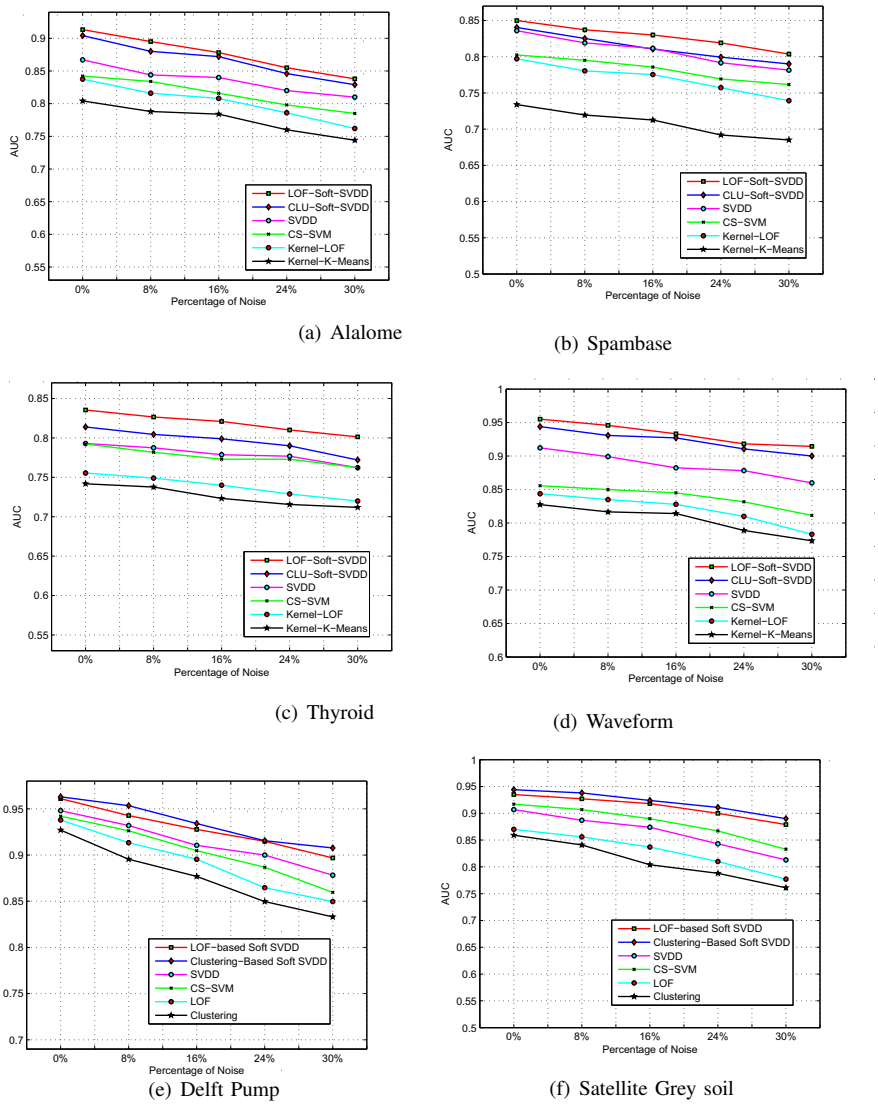
E. Impact of Imbalanced Data Distribution

So far we have demonstrated that our proposed method can consistently outperform CS-SVM when the number of abnormal data is much smaller than the number of normal data. However, it is still interesting to see how the performance of the three algorithms would be affected when more abnormal data are available for training.

Table II
COMPARISON OF AUC VALUES WITH RESPECT TO DIFFERENT RATIOS OF NORMAL DATA SIZE TO ABNORMAL DATA SIZE IN THE TRAINING DATASET

Datasets	Ratio	CLU-Soft-SVDD	LOF-Soft-SVDD	CS-SVM
Abalone	98:2	0.887	**0.892**	0.765
	95:5	0.904	**0.913**	0.842
	90:10	0.913	**0.918**	0.915
Spambase	98:2	0.835	**0.843**	0.746
	95:5	0.840	**0.850**	0.803
	90:10	0.844	**0.858**	0.853
Thyroid	98:2	0.804	**0.813**	0.786
	95:5	0.813	**0.836**	0.812
	90:10	0.825	**0.840**	0.836
Waveform	98:2	0.934	**0.939**	0.706
	95:5	0.944	**0.955**	0.856
	90:10	0.957	**0.966**	0.953
Delft Pump	98:2	0.946	**0.951**	0.826
	95:5	0.961	**0.963**	0.942
	95:10	0.968	**0.968**	0.961
Satellite Grey soil	95:2	0.924	**0.935**	0.807
	95:5	0.935	**0.944**	0.917
	90:10	0.938	**0.948**	0.944

Table II shows the AUC values with respect to different ratios of normal data size to abnormal data size in the training data. It is noted that as more abnormal examples are added into the training dataset, CS-SVM offers increasing



(a) Alalome (b) Spambase

(c) Thyroid (d) Waveform

(e) Delft Pump (f) Satellite Grey soil

Figure 4. Comparison of AUC values with respect to different percents of training data corrupted by noise

accuracy. This is because more negative examples can offer more information from negative class to build a more accurate SVM. However, when the ratio of normal data size to abnormal data size are $98:2$ and $95:5$ for which the number of abnormal examples are very few, LOF-Soft-SVDD and CLU-Soft-SVDD can remarkably outperform CS-SVM. This is because, based on insufficient abnormal data, CS-SVM cannot construct an accurate decision boundary to distinguish two classes. This indicates that, our proposed method can yield higher accuracy in real-world applications where abnormal data are very scarce.

V. Conclusion and Future Work

In this paper, we propose a new model-based approach to outlier detection by introducing a confidence value to each input data into the SVDD training phase. Our proposed method first captures the local uncertainty by computing a confidence value based on each example's local data behavior, and then builds a global classifier for outlier detection by extending the SVDD-based learning framework. Experiments on 10 real life datasets have shown that our proposed method can achieve a better tradeoff between detection rate and false alarm rate for outlier detection.

We plan to extend our work in several directions. First, we would like to investigate how to design better mechanisms to generate confidence values based on the data characteristics in a given application domain. Second, we will look into how to use an online process to learn the hypersphere boundary of Soft-SVDD in streaming environments.

VI. Acknowledgment

This work is sponsored in part by UTS QCIS, Australian Research Council through grants DP1096218, DP0988016, LP100200774 and LP0989721, and US NSF through grants IIS 0905215, DBI-0960443, OISE-0968341 and OIA-0963278.

References

[1] N. Abe, B. Zadrozny, and J. Langford. Outlier detection by active learning. In *KDD 2006*, pages 504–509, Philadelphia, USA, 2006.

[2] C. C. Aggarwal and P. S. Yu. Outlier detection with uncertain data. In *SDM 2008*, pages 483–493, Atlanta, Georgia, USA, 2008.

[3] F. Angiulli and C. Pizzuti. Outlier mining in large high-dimensional data sets. *TKDE*, 17(2):203–215, 2005.

[4] V. Barnett and T. Lewis. *Outliers in statistical data*. John Wiley, Chichester, 1994.

[5] J. Bi and T. Zhang. Support vector classification with input data uncertainty. In *NIPS 2004*, volume 17, pages 161–168, Vancouver, Canada, 2004.

[6] M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *SIGMOD 2000*, pages 93–104, Dallas, USA, May 2000.

[7] P. Chan and S. Stolfo. Toward scalable learning with non-uniform class and cost distributions. In *In KDD 1998*, pages 164–168, New York, NY, USA, 1998.

[8] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):Article 15, July 2009.

[9] C. Elkan. The foundations of cost-sensitive learning. In *IJCAI 2001*, pages 973–978, Seattle, USA, 2001.

[10] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *ICML 2000*, pages 255–262, San Francisco, CA, USA, June–July 2000.

[11] G. Fumera and F. Roli. Cost-sensitive learning in support vector machines. In *Proceedings of the Workshop on Machine Learning, Methods and Applications*, Siena, Italy, September 2002.

[12] A. Ghoting, S. Parthasarathy, and M. E. Otey. Fast mining of distance-based outliers in high-dimensional datasets. *DMKD*, 16(3):349–364, June 2008.

[13] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[14] S. Y. Jiang and Q.-B. An. Clustering-based outlier detectiion method. In *ICFSKD 2008*, pages 429–433, Jinan, Shandong, China, October 2008.

[15] E. M. Jordaan and G. F. Smits. Robust outlier detection using svm regression. In *IJCNN 2004*, pages 1098–7576, Budapest, Hungary, July 2004.

[16] U. Knoll, G. Nakhaeizadeh, and B. Tausend. Cost-sensitive pruning of decision trees. In *ECML 1994*, pages 383–386, Catania, Italy, April 1994.

[17] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *KDD 2005*, pages 157–166, Chicago, Illinois, USA, August 2005.

[18] C. Li and W. H. Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. In *Proceedings of the National Academy of Sciences USA*, volume 98, pages 31–36, January 2001.

[19] Y. Lin, Y. Lee, and G. Wahba. Support vector machine for classification in nonstandard situations. *Machine Learning*, 46:191–202, 2002.

[20] C. X. Ling, J. Huang, and H. Zhang. Auc: a statistically consistent and more discriminating measure than accuracy. In *In IJCAI 2003*, pages 519–526, Acapulco, Mexico, August 2003.

[21] Y. Batistakis M. Halkidi and M. Vazirgiannis. Cluster validity methods: Part I. *ACM SIGMOD Record*, 31:40–45, 2002.

[22] R. Smith, A. Bivens, M. Embrechts, C. Palagiri, and B. Szymanski. Clustering approaches for anomaly based intrusion detection. In *Proceedings of Intelligent Engineering Systems through Artificial Neural Networks*, pages 579–584. ASME Press, 2002.

[23] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *JMLR*, 6:211–232, 2005.

[24] David Tax and R. Duin. Outlier detection using classifier instability. In *Advances in Pattern Recognition, Lecture notes in Computer Science*, pages 593–601, 1998.

[25] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[26] D. M. J. Tax, A. Ypma, and R. P. W. Duin. Support vector data description applied to machine vibration analysis. In *ASCI 1999*, pages 398–405, Heijen, The Netherlands, June 1999.

[27] J. Theller and D.M. Cai. Resampling approach for anomaly detection in multispectral images. In *Proceedings of the SPIE*, volume 5093, pages 230–240, San Diego, CA, USA, August 2003.

[28] V. N. Vapnik. *Statistical learning theory*. John Wiley and Sons, 1998.

[29] M. Wu and J. Ye. A small sphere and large margin approach for novelty detection using training data with outliers. *IEEE TPAMI*, 31(11):2088–2092, 2009.