Active and Semi-supervised Data Domain Description

Nico Görnitz, Marius Kloft, and Ulf Brefeld

Machine Learning Group
Technische Universität Berlin
Franklinstr. 28/29, 10587 Berlin, Germany
{goernitz,mkloft,brefeld}@cs.tu-berlin.de

Abstract. Data domain description techniques aim at deriving concise descriptions of objects belonging to a category of interest. For instance, the support vector domain description (SVDD) learns a hypersphere enclosing the bulk of provided unlabeled data such that points lying outside of the ball are considered anomalous. However, relevant information such as expert and background knowledge remain unused in the unsupervised setting. In this paper, we rephrase data domain description as a semi-supervised learning task, that is, we propose a semi-supervised generalization of data domain description (SSSVDD) to process unlabeled *and* labeled examples. The corresponding optimization problem is non-convex. We translate it into an unconstraint, continuous problem that can be optimized accurately by gradient-based techniques. Furthermore, we devise an effective active learning strategy to query low-confidence observations. Our empirical evaluation on network intrusion detection and object recognition tasks shows that our SSSVDDs consistently outperform baseline methods in relevant learning settings.

1 Introduction

Data domain description techniques aim to devise concise descriptions of observed data. The task is to find minimal regions in feature space containing all data points that belong to the category of the observed data. Observations that do not fall into this region deviate from the normality and are rejected.

Data domain description techniques are therefore frequently being applied to outlier and anomaly detection problems where a model of normality is devised from available observations. Anomality of new objects is measured by their distance (in some metric space) from the learned model of normality, historically also known as "the sense of self" [7].

In network intrusion detection, the main merit of anomaly detection techniques is their ability to detect previously unknown attacks. One might think that the collective expertise amassed in the computer security community rules out major outbreaks of "genuinely novel" exploits. Unfortunately, a wide-scale deployment of efficient tools for obfuscation, polymorphic mutation and encryption results in an exploding variability of attacks. Although being only "marginally

novel", such attacks quite successfully defeat signature-based detection tools. This reality brings one-class anomaly detection back into the research focus of the security community [14, 15, 11, 28, 27, 19, 20]. Until now, anomaly detection is usually being regarded as an unsupervised learning task for good reasons: Firstly, the rejection class cannot be sampled per definition as it comprises rare and unlikely events. Secondly, outliers are frequently too diverse to be modeled by only a single rejection class. However, multi-class approaches to anomaly detection are also inappropriate because keeping track of changing class-distributions is intractable for real applications including spam filtering and network intrusion detection.

Nevertheless, data domain description techniques exhibit appealing properties for dealing with multiple, non-stationary class-distributions in settings where shifting distributions can be modeled by all means. For instance, domain descriptions have been successfully applied to one-class and multi-class classification problems with temporally varying numbers of categories such as anomaly and event detection and object recognition tasks. Instead of maintaining expensive multi-class classifiers that have to be retrained using all available data once a new category is added, one simply learns a single domain description for every (new) category of interest.

We claim that an unsupervised learning setting for data domain description is often too restricted for practical applications. Firstly, these methods have to be trained solely on normal data which is hardly possible without already *knowing* the labelings. Although state-of-the-art techniques prove robust against injecting a few instances of the rejection class into the training data [2, 24], knowing the class ratios is often crucial for accurate parameter adjustments. Secondly, one often knows the categories of certain training instances, be it manually labeled or recently seen instances. Such expert knowledge cannot be exploited in unsupervised settings and the learned models are sub-optimal in the sense that they leave out important information.

In this paper, we rephrase data domain description as a semi-supervised learning task, that is, we present semi-supervised data domain description (SSSVDD) that allows for processing unlabeled as well as labeled data to include expert and prior knowledge. Our model learns a minimal enclosing hypersphere in feature space that contains the normal data where point-wise errors are relaxed by slack variables. The inclusion of examples of the rejection class turns the optimization problem non-convex. As a remedy, we translate the optimization into an unconstraint, continuous problem with fewer parameters. It can therefore be optimized faster, and the retrieved local minima are substantially better on average [3]. The SSSVDD contains the unsupervised data domain description [24] as a special case that is obtained when no label information is used in the training process.

Furthermore, we devise an active learning strategy to query low-confidence decisions, hence guiding the user in the labeling process. Active learning selects an instance to be labeled by the user from the pool of unlabeled data. The selection process is designed to find unlabeled examples in the pool which –

once labeled – lead to the maximal improvement of the hypothesis. Thus, the SSSVDD is initially trained solely on unlabeled examples and then subsequently refined by incorporating labeled examples that have been queried by the active learning rule. The training process can be terminated at any time, for instance when the desired predictive performance is obtained.

Empirical results on network intrusion detection and object recognition tasks show the benefit of casting data domain description into a semi-supervised learning framework: The SSSVDD significantly outperforms appropriate baseline methods for all learning settings. This effect is significantly enhanced by active learning. Our active learning strategy not only reduces the manual labeling effort for the practitioner, it also allows for automatically identifying novel network attacks for the intrusion detection tasks.

Our paper is structured as follows. Section 2 reviews related work and Section 3 introduces the classical data domain description. We extend the latter to a semi-supervised learning method in Section 4 where we also discuss optimization issues. Section 5 introduces our active learning strategy and Section 6 reports on empirical results. Section 7 concludes.

2 Related Work

Data domain description is usually regarded as an unsupervised or one-class classification task. Prominent approaches comprise k-nearest neighbors [2] or other distance based methods [9], quadratic programming [24], and statistical methods [30, 25]. In this paper, we rephrase data domain description as a semi-supervised task (see [32] for an overview).

Active learning for anomaly detection has been studied by [22, 17, 1]. [1] take a max-margin approach and propose to query points that lie close to the decision hyperplane and violate the margin criterion in order to minimize the error rate. By contrast, the approach by [17] aims at detecting rejection categories in the data using as few queries as possible. Finally, the approach taken in [22] combines the former two active learning strategies to find interesting regions in feature space *and* to decrease the error-rate simultaneously.

Furthermore, there are several extensions of unsupervised data domain descriptors allowing for the inclusion of labeled examples. For instance, [8, 12, 26, 31] present fully-supervised variants of the classical support vector data description (SVDD) [24]. However, the objective functions are no longer convex and the proposed optimizations in dual space may suffer from duality gaps. Another variant proposed in [23] is trained on unlabeled and instances belonging to the rejection class. Although this approach seems promising, it also suffers from non-convexity of the objective.

3 Support Vector Data Description

In this section, we briefly review the classical support vector domain description (SVDD) [24]. We are given a set of n *normal* inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ and a

function $\phi : \mathcal{X} \to \mathcal{F}$ extracting features out of the inputs. For instance, $\boldsymbol{x}_i$ may refer to the i-th recorded request and $\phi(\boldsymbol{x}_i)$ may encode the vector of bigrams occurring in $\boldsymbol{x}_i$.

The goal of the SVDD is to find a concise description of the normal data such that anomalous data can be easily identified as outliers. In the underlying one-class scenario, this translates to finding a minimal enclosing hypersphere (i.e., center $\boldsymbol{c}$ and radius R) that contains the normal input data [24], see Figure 1 (left). Given the function

$$f(\boldsymbol{x}) = \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 - R^2,$$

the boundary of the ball is described by the set $\{\boldsymbol{x} : f(\boldsymbol{x}) = 0 \wedge \boldsymbol{x} \in \mathcal{X}\}$. That is, the parameters of f are to be chosen such that $f(\boldsymbol{x}) \leq 0$ for normal data and $f(\boldsymbol{x}) > 0$ for anomalous points. The center $\boldsymbol{c}$ and the radius R can be computed accordingly by solving the following optimization problem [24]

$$\begin{aligned} \min_{R,\boldsymbol{c},\boldsymbol{\xi}} \quad & R^2 + \eta \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \forall_{i=1}^{n} : \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \leq R^2 + \xi_i \\ & \forall_{i=1}^{n} : \xi_i \geq 0. \end{aligned} \tag{1}$$

The trade-off parameter η adjusts point-wise violations of the hypersphere. That is, a concise description of the data might benefit from omitting some data points in the computation of the solution. Discarded data points induce slack that is absorbed by variables ξ_i. Thus, in the limit $\eta \to \infty$, the hypersphere will contain all input examples irrespectively of their utility for the model and $\eta \to 0$ implies $R \to 0$ and the center $\boldsymbol{c}$ reduces to the centroid of the data.

The above optimization problem can be translated into an equivalent dual formulation by exploiting the identity $\boldsymbol{c} = \sum_{i=1}^{n} \alpha_i \phi(\boldsymbol{x}_i)$. We arrive at the dual SVDD optimization problem [24],

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n} \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i = 1 \quad \text{and} \quad 0 \leq \alpha_i \leq \eta \quad \forall i = 1, \ldots, n. \end{aligned}$$

Once optimal parameters $\boldsymbol{\alpha}^*$ are found these are used as plug-in estimates to compute the anomaly score for new and unseen instances. A new observation $\bar{\boldsymbol{x}}$ is accepted if

$$k(\bar{\boldsymbol{x}}, \bar{\boldsymbol{x}}) - 2 \sum_{i=1}^{n} \alpha_i^* k(\boldsymbol{x}_i, \bar{\boldsymbol{x}}) + \sum_{i,j=1}^{n} \alpha_i^* \alpha_j^* k(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq R^2.$$

[8, 12, 26, 23] propose extensions of the SVDD to incorporate labeled data into the learning process. The corresponding optimization problems are however not convex and the dual solution might suffer from a duality gap.

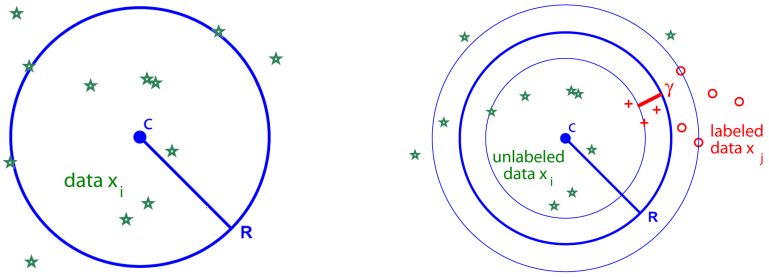

Fig. 1. Left: An exemplary solution of the SVDD. Right: Illustration of SSSVDD that incorporates unlabeled (green) as well as labeled data of the normal class (red) and the rejection category (blue).

4 Semi-supervised Data Domain Description

In this section, we derive our semi-supervised data domain description. In addition to n normal observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ we are also given m labeled pairs $(\boldsymbol{x}^*_{n+1}, y^*_{n+1}), \ldots, (\boldsymbol{x}^*_{n+m}, y^*_{n+m}) \subset \mathcal{X} \times \{+1, -1\}$, where we associate normal data with the positive class and outliers as the negative class. As in the previous section, we aim at finding a model $f(\boldsymbol{x}) = ||\phi(\boldsymbol{x}) - \boldsymbol{c}||^2 - R^2$ that generalizes well on unseen data, however, the model is now devised on the basis of labeled and unlabeled data. A straight-forward extension of the SVDD in Equation (1) using both, labeled and unlabeled examples, is given by

$$
\begin{aligned}
\min_{R,\gamma,\boldsymbol{c},\boldsymbol{\xi}} \quad & R^2 - \kappa\gamma + \eta_u \sum_{i=1}^{n} \xi_i + \eta_l \sum_{j=n+1}^{n+m} \xi^*_j \\
\text{s.t.} \quad & \forall_{i=1}^{n} : \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \leq R^2 + \xi_i \\
& \forall_{j=n+1}^{n+m} : y^*_j \left(\|\phi(\boldsymbol{x}^*_j) - \boldsymbol{c}\|^2 - R^2\right) \leq -\gamma + \xi^*_j \\
& \forall_{i=1}^{n} : \xi_i \geq 0, \\
& \forall_{j=n+1}^{n+m} : \xi^*_j \geq 0.
\end{aligned}
\tag{2}
$$

The optimization problem has additional constraints for the labeled examples that have to fulfill the margin criterion with margin γ. Trade-off parameters κ, η_u, and η_l balance margin-maximization and the impact of unlabeled and labeled examples, respectively. To avoid cluttering the notation unnecessarily, we omit the obvious generalization of allowing different trade-offs η_l^+ and η_l^- for positively and negatively labeled instances, respectively. The additional slack variables ξ^*_j are bound to labeled examples and allow for point-wise relaxations of margin violations by labeled examples. The solution of the above optimization problem is illustrated in Figure 1 (right).

The inclusion of negatively labeled data turns the above optimization problem non-convex and optimization in the dual is prohibitive. As a remedy, we

translate Equation (2) into an unconstraint, continuous problem [3, 33]. For the above problem, it is possible to resolve the slack terms:

$$\xi_i = \ell\left(R^2 - ||\phi(\boldsymbol{x}_i) - \boldsymbol{c}||^2\right)$$
$$\xi_j^* = \ell\left(y_j^*\left(R^2 - ||\phi(\boldsymbol{x}_j^*) - \boldsymbol{c}||^2\right) - \gamma\right)$$

where $\ell(t) = \max\{-t, 0\}$ is the common hinge loss where we explicitely deal with the margin γ in the argument t because γ is part of the optimization. We can now pose optimization problem (2) as a simple minimization problem *without* constraints as follows,

$$\begin{aligned}\min_{R,\gamma,\boldsymbol{c}} \quad & R^2 - \kappa\gamma + \eta_u \sum_{i=1}^{n} \ell\left(R^2 - ||\phi(\boldsymbol{x}_i) - \boldsymbol{c}||^2\right) \\ & + \eta_l \sum_{j=n+1}^{n+m} \ell\left(y_j^*\left(R^2 - ||\phi(\boldsymbol{x}_j^*) - \boldsymbol{c}||^2\right) - \gamma\right). \end{aligned} \tag{3}$$

Note that the optimization problems in Equations (2) and (3) are equivalent so far. We now substitute the Huber loss for the hinge loss to obtain a smooth and differentiable function that can be optimized with gradient-based techniques. The Huber loss $\ell_{\Delta,\epsilon}$ is displayed in Figure 2 and given by

$$\ell_{\Delta,\epsilon}(t) = \begin{cases} \Delta - t & : \quad t \leq \Delta - \epsilon \\ \frac{(\Delta+\epsilon-t)^2}{4\epsilon} & : \quad \Delta - \epsilon \leq t \leq \Delta + \epsilon \\ 0 & : \quad \text{otherwise} \end{cases}$$
$$\ell'_{\Delta,\epsilon}(t) = \begin{cases} -1 & : \quad t \leq \Delta - \epsilon \\ -\frac{1}{2}(\frac{\Delta-t}{\epsilon} + 1) & : \quad \Delta - \epsilon \leq t \leq \Delta + \epsilon \\ 0 & : \quad \text{otherwise}\,. \end{cases} \tag{4}$$

For notational convenience, we focus on the Huber loss for $\ell_{\Delta=0,\epsilon}(t)$ and move margin dependent terms into the argument t. Using the Huber loss $\ell_{0,\epsilon}$, computing the gradients of the slack variables ξ_i associated with unlabeled examples with respect to the primal variables R and $\boldsymbol{c}$ yields

$$\frac{\partial \xi_i}{\partial R} = 2R\ell'_\epsilon(R^2 - ||\phi(\boldsymbol{x}_i) - \boldsymbol{c}||^2)$$
$$\frac{\partial \xi_i}{\partial \boldsymbol{c}} = 2(\phi(\boldsymbol{x}_i) - \boldsymbol{c})\ell'_\epsilon(R^2 - ||\phi(\boldsymbol{x}_i) - \boldsymbol{c}||^2).$$

The derivatives of their counterparts ξ_j^* for the labeled examples with respect to R, γ, and $\boldsymbol{c}$ are given by

$$\frac{\partial \xi_j^*}{\partial R} = 2y_j^* R\ell'_\epsilon\left(y_j^*\left(R^2 - ||\phi(\boldsymbol{x}_j^*) - \boldsymbol{c}||^2\right) - \gamma\right)$$
$$\frac{\partial \xi_j^*}{\partial \gamma} = -\ell'_\epsilon\left(y_j^*\left(R^2 - ||\phi(\boldsymbol{x}_j^*) - \boldsymbol{c}||^2\right) - \gamma\right)$$
$$\frac{\partial \xi_j^*}{\partial \boldsymbol{c}} = 2y_j^*(\phi(\boldsymbol{x}_j^*) - \boldsymbol{c})\ell'_\epsilon\left(y_j^*\left(R^2 - ||\phi(\boldsymbol{x}_j^*) - \boldsymbol{c}||^2\right) - \gamma\right).$$

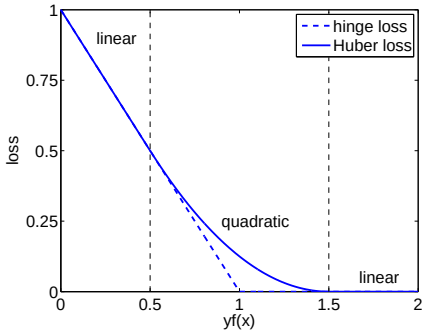

Fig. 2. The differentiable Huber loss $\ell_{\Delta=1,\epsilon=0.5}$.

Substituting the partial gradients, we resolve the gradient of Equation (3) with respect to the primal variables:

$$\frac{\partial EQ3}{\partial R} = 2R + \eta_u \sum_{i=1}^{n} \frac{\partial \xi_i}{\partial R} + \eta_l \sum_{j=n+1}^{n+m} \frac{\partial \xi_j^*}{\partial R}, \tag{5}$$

$$\frac{\partial EQ3}{\partial \gamma} = -\kappa + \eta_l \sum_{j=n+1}^{n+m} \frac{\partial \xi_j^*}{\partial \gamma}, \tag{6}$$

$$\frac{\partial EQ3}{\partial \boldsymbol{c}} = \eta_u \sum_{i=1}^{n} \frac{\partial \xi_i}{\partial \boldsymbol{c}} + \eta_l \sum_{j=n+1}^{n+m} \frac{\partial \xi_j^*}{\partial \boldsymbol{c}}. \tag{7}$$

The above equations can be plugged directly into off-the-shelf gradient-based optimization tools to optimize Equation (3) in the input space for the identity $\phi(\boldsymbol{x}) = \boldsymbol{x}$. However, predictive power is often related to (possibly) non-linear mappings ϕ of the input data into some high-dimensional feature space. In the following, we extend our approach to allow for the use of kernel functions. An application of the representer theorem (see Appendix) shows that the center $\boldsymbol{c}$ can be expanded as

$$\boldsymbol{c} = \sum_i \alpha_i \phi(\boldsymbol{x}_i) + \sum_j \alpha_j y_j^* \phi(\boldsymbol{x}_j^*). \tag{8}$$

According to the chain rule, the gradient of Equation (3) with respect to the $\alpha_{i/j}$ is given by

$$\frac{\partial EQ3}{\partial \alpha_{i/j}} = \frac{\partial EQ3}{\partial \boldsymbol{c}} \frac{\partial \boldsymbol{c}}{\partial \alpha_{i/j}}.$$

Using Equation (8), the partial derivatives $\frac{\partial \boldsymbol{c}}{\partial \alpha_{i/j}}$ resolve to

$$\frac{\partial \boldsymbol{c}}{\partial \alpha_i} = \phi(\boldsymbol{x}_i) \quad \text{and} \quad \frac{\partial \boldsymbol{c}}{\partial \alpha_j} = y_j^* \phi(\boldsymbol{x}_j^*), \tag{9}$$

respectively. Applying the cain-rule to Equations (5),(6),(7), and (9) gives the gradients of Equation (3) with respect to the $\alpha_{i/j}$. The final objective function allowing for the use of kernel functions can be stated as

$$\begin{aligned}\min_{R,\gamma,\alpha} \quad & R^2 - \kappa\gamma + \eta_u \sum_{i=1}^{n} \ell_\epsilon \left(R^2 - k(\boldsymbol{x}_i, \boldsymbol{x}_i) + (2\boldsymbol{e}_i - \alpha)' K \alpha\right) \\ & + \eta_l \sum_{j=n+1}^{n+m} \ell_\epsilon \left(y_j^* \left(R^2 - k(\boldsymbol{x}_j^*, \boldsymbol{x}_j^*) + (2\boldsymbol{e}_j^* - \alpha)' K \alpha\right) - \gamma\right), \quad (10)\end{aligned}$$

where kernel K is given by $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$ and $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{n+m}$ is the standard base of $\mathbb{R}^{n+m}$. By rephrasing the problem as an unconstrained optimization problem, its intrinsic complexity has not changed. However, the local minima of Optimization Problems (3) and (10) can now easily be found with gradient-based techniques such as conjugate gradient descent. In general, unconstrained optimization is also easier to implement than constrained optimization. We will observe the benefit of this approach in the following.

5 Active Learning

The SSSVDD is initially trained solely on unlabeled examples and then subsequently refined by incorporating labeled examples that have been queried by the active learning rule. We now devise an active learning strategy to query low-confidence decisions, hence guiding the user in the labeling process. Our active learning strategy selects an instance of the unlabeled data pool to be labeled by the user. The selection process is designed to find the unlabeled example in the pool which – once labeled – leads to the maximal improvement of the actual model.

We begin with a commonly used active learning strategy which simply queries borderline points. The strategy is sometimes called *margin strategy* and can be expressed by asking the user to label the point $\boldsymbol{x}'$ that is closest to the decision hypersphere [1, 29]

$$\boldsymbol{x}' = \operatorname*{argmin}_{\boldsymbol{x} \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}} \frac{|f(\boldsymbol{x})|}{\Omega} = \operatorname*{argmin}_{\boldsymbol{x} \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}} \frac{|R^2 - \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2|}{\Omega}, \quad (11)$$

where Ω is a normalization constant and given by $\Omega = \max_i |f(\boldsymbol{x}_i)|$.

However, when dealing with many non-stationary outlier and/or attack categories, it is beneficial to identify novel reject classes as soon as possible. We translate this into an active learning strategy as follows. Let $A = (a_{ij})_{i,j=1,\ldots,n+m}$ be an adjacency matrix, for instance obtained by a k-nearest-neighbor approach, where $a_{ij} = 1$ if $\boldsymbol{x}_i$ is among the k-nearest neighbors of $\boldsymbol{x}_j$ and 0 otherwise. We introduce an extended labeling $\bar{y}_1 \ldots, \bar{y}_{n+m}$ for all examples by defining $\bar{y}_i = 0$ for unlabeled instances and retaining the labels for labeled intances, i.e.,

$\bar{y}_j = y_j$. Using these pseudo labels, Equation (12) returns the unlabeled instance according to

$$\boldsymbol{x}' = \underset{\boldsymbol{x}_i \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}}{\operatorname{argmin}} \frac{1}{2k} \sum_{j=1}^{n+m} (\bar{y}_j + 1)\, a_{ij}. \tag{12}$$

The above strategy explores unknown regions in feature space and subsequently deepens the learned knowledge by querying clusters of potentially similar objects to allow for good generalizations.

Nevertheless, using Equation (12) alone may result in querying points lying close to the center of the hypersphere or far from its boundary. These points will hardly contribute to an improvement of the hypersphere. In other words, only a combination of both strategies (11) and (12) guarantees the active learning to query points of interest. Our final active learning strategy is therefore given by

$$\boldsymbol{x}' = \underset{\boldsymbol{x}_i \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}}{\operatorname{argmin}} = \delta \frac{|f(\boldsymbol{x})|}{\Omega} + \frac{1-\delta}{2k} \sum_{j=1}^{n+m} (\bar{y}_j + 1)\, a_{ij} \tag{13}$$

for $\delta \in [0, 1]$. The combined strategy queries instances that are close to the boundary of the hypersphere *and* lie in potentially anomalous clusters with respect to the k-nearest neighbor graph. Depending on the actual value of δ, the strategy jumps from cluster to cluster and thus helps to identify interesting regions in feature space. For the special case of no labeled points our combined strategy reduces to the margin strategy.

Usually, an active learning step is followed by an optimization step of the SSSVDD taking into account the newly labeled data. This procedure is of course time-consuming and can be altered for practical settings, for instance by querying a couple of points before performing a model update. Irrespectively of the actual implementation, alternating between active learning and updating the model can be repeated until a desired predictive performance is obtained.

6 Empirical Results

In this section, we empirically evaluate the SSSVDD and the active learning strategies and compare their performances to appropriate strawmen. The baselines SVDD and SVDD^{neg} [23] are implemented in Matlab and optimized by SMO [18]. Additional baselines for the object recognition tasks are binary SVMs. SSSVDDs are optimized by conjugate gradient descent. Parameters of the active learning strategy are set to $k = 10$, $\alpha = 0.1$ for simplicity. We experiment on network intrusion and object recognition tasks.

6.1 Intrusion Detection

For the intrusion detection experiments we use HTTP traffic recorded within 10 days at Fraunhofer Institute FIRST. The data set comprises 145,069 unmodified

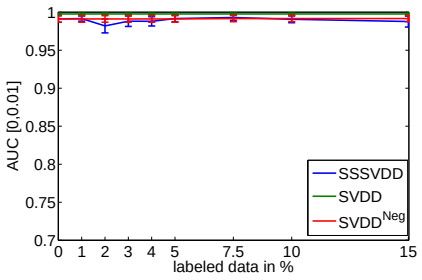

Fig. 3. Results for normal vs. malicious.

connections of average length of 489 bytes. We refer to the FIRST data as the *normal pool*. The *malicious pool* contains 27 real attack classes generated using the Metasploit framework [16]. It covers 15 buffer overflows, 8 code injections and 4 other attacks including HTTP tunnels and cross-site scripting. Every attack is recorded in $2-6$ different variants using virtual network environments and decoy HTTP servers.

To study the robustness of the different approaches in a more realistic scenario we also study techniques to obfuscate malicious content by adapting attack payloads to mimic benign traffic in feature space [6]. As a consequence, the extracted features do not deviate from a model of normality and the classifier is likely to be fooled by the attack. For our purposes, it already suffices to study a simple cloaking technique by adding common HTTP headers to the payload while the malicious body of the attack remains unaltered. We apply this technique to the malicious pool and refer to the obfuscated set of attacks as *cloaked pool*.

We focus on two scenarios: normal vs. malicious and normal vs. cloaked data. For both settings, the respective byte streams are translated into a bag-of-3-grams representation. For each experiment, we randomly draw 966 training examples from the normal pool and 34 attacks either from the malicious or the cloaked pool, depending on the scenario. Holdout and test sets are also drawn at random and consist of 795 normal connections and 27 attacks, each. We make sure that attacks of the same attack class occur either in the training, or in the test set but not in both. We report on 10 repetitions with distinct training, holdout, and test sets and measure the performance by the area under the ROC curve in the false-positive interval $[0, 0.01]$ ($\text{AUC}_{0.01}$).

Figure 3 shows the results for normal vs. malicious data pools, where the x-axis depicts the percentage of randomly drawn labeled instances. Irrespectively of the amount of labeled data, the malicious traffic is detected by all methods equally well as the intrinsic nature of the attacks is well captured by the bag-of-3-grams representation. There is no significant difference between the classifiers. However, our next experiment shows the fragility of these results in the presence of simple cloaking techniques. Simply obfuscating the attacks by copying normal headers into the malicious payload leads to dramatically different results.

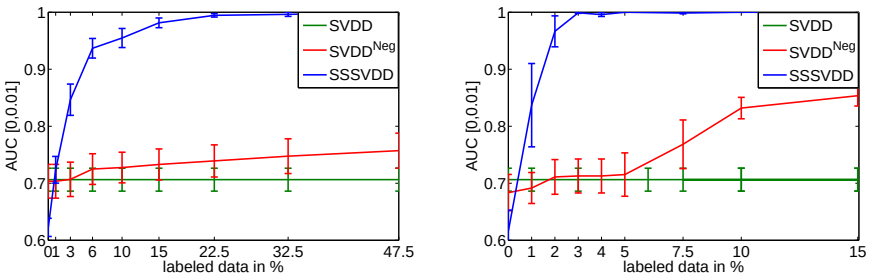

Fig. 4. Results for normal vs. cloaked. Left: Random samling. Right: Active learning.

Figure 4 (left) displays the results for normal vs. cloaked data. First of all, the performance of the unsupervised SVDD drops to only 70%. We obtain a similar result for the SVDD^{neg}; incorporating cloaked attack information into the training process of the SVDD leads to an increase of about 5% which is far from any practical value. Notice that the SVDD^{neg} cannot make use of labeled data of the normal class. Thus, its moderate ascent in terms of the number of labeled examples is credited to the class ratio of 966/34 for the random labeling strategy. The bulk of additional information cannot be exploited and has to be left out. By contrast, the semi-supervised SSSVDD includes all labeled data into the training process and clearly outperforms the two baselines. For only 5% labeled data, the SSSVDD easily beats the best baseline and for randomly labeling 30% of the available data it separates almost perfectly between normal and cloaked malicious traffic.

Nevertheless, labeling 30% of the data is not realistic for practical applications. We thus explore the benefit of active learning for inquiring label information of borderline and low-confidence points. Figure 4 (right) shows the results for normal vs. cloaked data where the labeled data for SVDD^{neg} and SSSVDD is chosen according to the active learning strategy in Equation (13). The unsupervised SVDD that does not make use of labeled information remains at an $\text{AUC}_{0.01}$ of 70%. Compared to the results for a random labeling strategy (Figure 4, left), the performance of its counterpart SVDD^{neg} increases. The ascent of the SVDD^{neg} is now steeper and yields 85% for 15% labeled data. However, the SSSVDD also improves for active learning and dominates the baselines. Using active learning, we need to label only 3% of the data for attaining an almost perfect separation, compared to 25% for a random labeling strategy. Our active learning strategy effectively boosts the performance and reduces the manual labeling effort significantly.

Figure 5 details the impact of our active learning strategy in Equation (13). We compare the number of outliers detected by the combined strategy with the margin-based strategy in Equation (11) (see also [1]) and by randomly drawing instances from the unlabeled pool. As a sanity check, we also included the theoretical outcome for random sampling. The results show that the combined

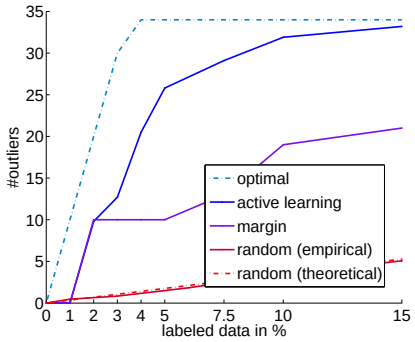

Fig. 5. Number of outliers found by active learning.

strategy effectively detects malicious traffic much faster than the margin-based strategy.

6.2 Object Recognition

For our object recognition experiments we use the classification data of the VOC 2008 challenge [5]. The data set comprises 8780 images and 20 object classes. An image is annotated with a class label if at least one object from that class is detectable in the image. We use the training and holdout sets for our experiments which contain 4340 images.

Fig. 6. Exemplary images from the VOC2008 object recognition data set. From left to right: aeroplane, dog, and bird.

We focus on the three randomly drawn classes *aeroplane* (198 instances), *bird* (286 instances), and *dog* (266 instances), exemplary images are displayed in Figure 6. From the pool, we draw 375 instances randomly as independent test set while the remaining 375 examples are used for model selection over 10 repetitions. In each run, we randomly draw 10 labeled images of each class, 148 unlabeled instances, and 187 holdout examples.

We employ pyramid histograms [10] of visual words [4] (PHOW) for pyramid levels 0,1,2 over the grey channel. We obtain a feature vector for every image by concatenating histograms of all levels. For the grey channel, 1200 visual words are computed by k-means clustering on SIFT features [13] from randomly drawn

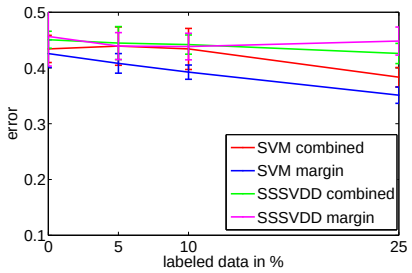

Fig. 7. Error rates for the VOC2008 data set.

images of each class. The underlying SIFT features are extracted from a dense grid of pitch ten.

Figure 7 compares regular support vector machines (SVMs) with SSSVDDs where both approaches apply margin-based active learning (Equation (11)) and the combined strategy in Equation (13) for detecting query points. For only a few labeled data points and many unlabeled examples (which cannot be utilized by SVMs), both approaches perform comparably. However, for increasing percentages of labeled data, the task becomes more and more a binary problem for which the SVM is well suited. For 25% labeled data, the SVM beats the SSSVDD clearly. Nevertheless, SSSVDD proves robust when labeled data is scarce and expensive to obtain; unlabeled examples are effectively exploited to augment sparse labelings.

7 Conclusion

In this paper, we proposed to view data domain description as a semi-supervised learning problem to allow for the inclusion of prior and expert knowledge. We generalized support vector data description to a semi-supervised learning algorithm (SSSVDD). Since the objective function of the SSSVDD is not convex, we translated the optimization problem into an unconstraint, continuous problem which can be optimized with efficient gradient-based techniques. Furthermore, we proposed a novel active learning strategy to guide the user in the labeling process of the unlabeled data by querying instances that are not only close to the boundary of the hypersphere but also likely members of novel rejection categories.

Empirically, we showed on network intrusion detection and object recognition tasks that rephrasing the unsupervised problem setting as a semi-supervised task is worth the effort. For instance in the network intrusion detection task, SSSVDDs prove robust in scenarios where the performance of baseline approaches deteriorate due to obfuscation techniques. Moreover, we observe the effectiveness of our active learning strategy which significantly improves the quality of the SSSVDD and spares practitioners from labeling unnecessarily many data points.

Acknowledgements We thank Konrad Rieck for providing the kernels for the HTTP traffic and Christina Müller and Shinichi Nakajima for helping us with the object recognition task. This work was supported in part by the German Bundesministerium für Bildung und Forschung (BMBF) under the project ReMIND (FKZ 01-IS07007A) and by the FP7-ICT Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-216886.

References

1. M. Almgren and E. Jonsson. Using active learning in intrusion detection. *Proc. IEEE Computer Security Foundation Workshop*, 2004.
2. F. Angiulli. Condensed nearest neighbor data domain description. In *Advances in Intelligent Data Analysis VI*, 2005.
3. O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the International Workshop on AI and Statistics*, 2005.
4. G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, Prague, Czech Republic, May 2004.
5. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/, 2008.
6. P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee. Polymorphic blending attacks. In *Proceedings of USENIX Security Symposium*, 2006.
7. S. Forrest, S.A. Hofmeyr, A. Somayaji, and T.A. Longstaff. A sense of self for unix processes. In *Proc. of IEEE Symposium on Security and Privacy*, pages 120–128, Oakland, CA, USA, 1996.
8. Chu-Hong Hoi, Chi-Hang Chan, Kaizhu Huang, Michael Lyu, and Irwin King. Support vector machines for class representation and discrimination. In *Proceedings of the International Joint Conference on Neural Networks*, 2003.
9. E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, 1998.
10. S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, New York, USA, June 2006.
11. W. Lee and S.J. Stolfo. A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information Systems Security*, 3:227–261, 2000.
12. Y. Liu and Y. F. Zheng. Minimum enclosing and maximum excluding machine for pattern description and discrimination. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 129–132, Washington, DC, USA, 2006. IEEE Computer Society.
13. D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
14. M.V. Mahoney and P.K. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 376–385, 2002.

15. M.V. Mahoney and P.K. Chan. Learning rules for anomaly detection of hostile network traffic. In *Proc. of International Conference on Data Mining (ICDM)*, 2003.
16. K. Maynor, K. Mookhey, J. F. R. Cervini, and K. Beaver. Metasploit toolkit. In *Syngress*, 2007.
17. D. Pelleg and A. Moore. Active learning for anomaly and rare-category detection. *Proc. Advances in Neural Information Processing Systems*, 2004.
18. J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: support vector learning*, 1999.
19. Konrad Rieck and Pavel Laskov. Detecting unknown network attacks using language models. In *Detection of Intrusions and Malware, and Vulnerability Assessment, Proc. of 3rd DIMVA Conference*, LNCS, pages 74–90, July 2006.
20. Konrad Rieck and Pavel Laskov. Language models for detection of unknown attacks in network traffic. *Journal in Computer Virology*, 2(4):243–256, 2007.
21. B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
22. J. W. Stokes and J. C. Platt. Aladin: Active learning of anomalies to detect intrusion. Technical report, Microsoft Research, 2008.
23. D. M. J. Tax. *One-class classification*. PhD thesis, Technical University Delft, 2001.
24. David M.J. Tax and Robert P.W. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004.
25. M. Thottan and Chuanyi Ji. Anomaly detection in ip networks. *IEEE Transactions on Signal Processing*, 51(8):2191–2204, 2003.
26. J. Wang, P. Neskovic, and L. N. Cooper. Pattern classification via single spheres. *Computer Science: Discovery Science (DS)*, 2005.
27. K. Wang, J.J. Parekh, and S.J. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Adances in Intrusion Detection (RAID)*, pages 226–248, 2006.
28. K. Wang and S.J. Stolfo. Anomalous payload-based network intrusion detection. In *Recent Adances in Intrusion Detection (RAID)*, pages 203–222, 2004.
29. M. K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen. Active learning with support vector machines in the drug discovery process. *Journal of Chemical Information and Computer Sciences*, 43(2):667–673, 2003.
30. Dit yan Yeung and Calvin Chow. Parzen-window network intrusion detectors. In *In Proceedings of the Sixteenth International Conference on Pattern Recognition*, pages 385–388, 2002.
31. C. Yuan and D. Casasent. Pseudo relevance feedback with biased support vector machine. In *Proceedings of the International Joint Conference on Neural Networks*, 2004.
32. X. Zhu. Semi–supervised learning in literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.
33. A. Zien, U. Brefeld, and T. Scheffer. Transductive support vector machines for structured variables. In *Proceedings of the International Conference on Machine Learning*, 2007.

Appendix

In this section, we show the applicability of the representer theorem for semi-supervised support vector domain descriptions.

Theorem 1 (Representer Theorem [21]). *Let $\mathcal{H}$ be a reproducing kernel Hilbert space with a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, a symmetric positive semi-definite function on the compact domain. For any function $L : \mathcal{R}^n \to \mathbb{R}$, any nondecreasing function $\Omega : \mathbb{R} \to \mathbb{R}$. If*

$$J^* := \min_{f \in \mathcal{H}} \; J(f) := \min_{f \in \mathcal{H}} \; \Omega\left(||f||_{\mathcal{H}}^2\right) + L\left(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)\right)$$

is well-defined, then there exist $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$, such that

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \cdot) \tag{14}$$

achieves $J(f) = J^*$. Furthermore, if Ω is increasing, then each minimizer of $J(f)$ can be expressed in the form of Eq. (14).

Lemma 1. *The representer theorem can be applied to Equation* (3).

Proof. Recall the primal SSSVDD objective function which is given by

$$\begin{aligned} J(R, \gamma, \boldsymbol{c}) =& R^2 - \kappa\gamma + \eta_u \sum_{i=1}^{n} \ell\left(R^2 - ||\phi(\boldsymbol{x}_i) - \boldsymbol{c}||^2\right) \\ &+ \eta_l \sum_{j=n+1}^{n+m} \ell\left(y_j^* \left(R^2 - ||\phi(\boldsymbol{x}_j^*) - \boldsymbol{c}||^2\right) - \gamma\right). \end{aligned}$$

Substituting $T := R^2 - ||c||^2$ leads to the new objective function

$$\begin{aligned} J(T, \gamma, \boldsymbol{c}) =& ||c||^2 + T - \kappa\gamma + \eta_u \sum_{i=1}^{n} \ell\left(T - ||\phi(\boldsymbol{x}_i)||^2 + 2\phi(\boldsymbol{x}_i)'\boldsymbol{c}\right) \\ &+ \eta_l \sum_{j=n+1}^{n+m} \ell\left(y_j^* \left(T - ||\phi(\boldsymbol{x}_j^*)||^2 + 2\phi(\boldsymbol{x}_j^*)'\boldsymbol{c}\right) - \gamma\right). \end{aligned}$$

Expanding the center $\boldsymbol{c}$ in terms of labeled and unlabeled input examples is now covered by the representer theorem. After the optimization, T can be easily re-substituted to obtain the primal variables R, γ, and $\boldsymbol{c}$. □