Active Learning for Network Intrusion Detection

Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld
Technische Universität Berlin
Machine Learning Group
Franklinstr. 28/29
10587 Berlin, Germany
{goernitz,mkloft,rieck,brefeld}@cs.tu-berlin.de

ABSTRACT

Anomaly detection for network intrusion detection is usually considered an unsupervised task. Prominent techniques, such as one-class support vector machines, learn a hypersphere enclosing network data, mapped to a vector space, such that points outside of the ball are considered anomalous. However, this setup ignores relevant information such as expert and background knowledge. In this paper, we rephrase anomaly detection as an active learning task. We propose an effective active learning strategy to query low-confidence observations and to expand the data basis with minimal labeling effort. Our empirical evaluation on network intrusion detection shows that our approach consistently outperforms existing methods in relevant scenarios.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; I.2.6 [**Artificial Intelligence**]: Learning—*Parameter learning*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*

General Terms

Algorithms, Experimentation, Security

Keywords

Machine learning, anomaly detection, support vector data description, active learning, intrusion detection, network security

1. INTRODUCTION

Computer systems linked to the Internet are exposed to a plethora of network attacks and malicious code. Several threats, ranging from zero-day exploits to Internet worms, target network hosts every day; networked systems are generally at risk to be remotely compromised and misused for illegal purposes. While early attacks have been developed rather for fun than for profit, proliferation of current network attacks is driven by a criminal underground economy. Compromised systems are often misused for monetary gains including the distribution of spam messages and theft of confidential data. The success of these illegal businesses poses a severe threat to the security of network infrastructures. Alarming reports on an expanding dissemination of advanced attacks render sophisticated security systems indispensible, e.g. [13, 27].

Conventional defenses against such network threats rest on the concept of misuse detection. That is, attacks are identified in network traffic using known patterns of misuse, so-called attack signatures. While misuse detection effectively protects from known threats, it increasingly fails to cope with the amount and diversity of attacks. The time span required for crafting a signature from a newly discovered attack is insufficient for protecting from rapidly propagating malicious code, e.g. [14, 24]. Moreover, polymorphism employed in recent attacks obstructs modelling accurate signatures [25], such that there is a demand for alternative techniques for detection of attacks during their initial propagation.

Anomaly detection methods provide means for identifying unknown and novel attacks in network traffic and thereby complement regular security defenses, e.g. [11, 10, 8, 19, 6, 21]. Anomaly detection methods proceed by learning a model of normal network data and identifying unusual contents and potential attacks as deviations thereof – irrespective of the employed intrusion techniques. Although anomaly detection methods enable tracking novel threats, their practical deployment poses a dilemma to the security practitioner. On the one hand, recent attacks and network traffic are required to properly calibrate and validate a learning method during operation. On the other hand, providing labels for network traces on a regular basis renders application of learning methods intractable in practice. Unfortunately, calibrating a method using unlabeled data only is not an option either, as the learned model of normality may be easily foiled by adversarial traffic [17, 7].

In this paper, we consider payload-based anomaly detection methods, such as PAYL [32], Anagram [31] and McPAD [16], which model normality by mapping network payloads to a vector space and enclosing the resulting vectors in a hypersphere. In contrast to previous approaches, however, we phrase anomaly detection as an *active learning task*.

AISec'09, November 9, 2009, Chicago, Illinois, USA.

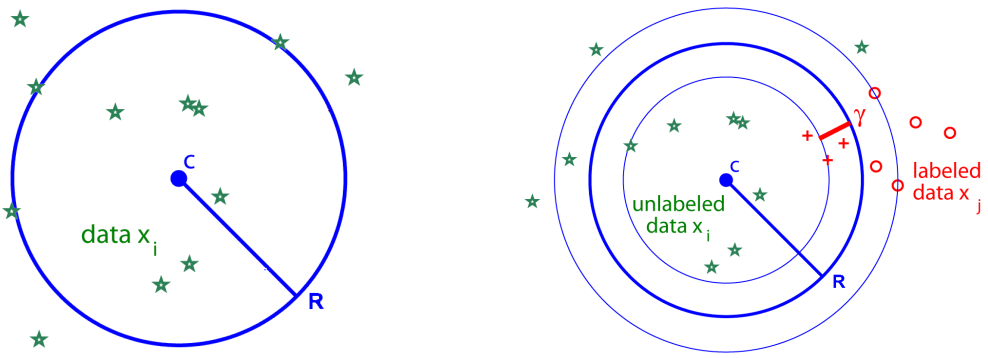

Figure 1: Left: An exemplary solution of the SVDD. Right: Illustration of ActiveSVDD that incorporates unlabeled (green) as well as labeled data of the normal class (red) and attacks (blue).

That is, we present a learning method that processes unlabeled data but actively queries labels for particular network payloads. The selection process is designed to find unlabeled examples in the data which – once labeled – lead to the maximal improvement of the learned hypersphere; the labeling effort for the practitioner is hereby significantly reduced.

We first present an effective active learning strategy to query network events of low confidence. The strategy calibrates the threshold of the hypersphere-based learner. Secondly, we extend hypersphere-based approaches to so-called *semi-supervised* models that enable processing unlabeled as well as labeled examples. Our method is initially trained on unlabeled examples and then subsequently refined by incorporating labeled data that have been queried by active learning rules. The training process can be terminated at any time, for instance when the desired predictive performance is obtained. The devised method contains unsupervised approaches such as centroids [32, 16, 21], as a special case that is obtained when no label information is used.

Empirical results on network intrusion detection demonstrate the benefit of combining anomaly detection and active learning. The active learning strategy significantly reduces the manual labeling effort for the practitioner. By labeling only a fraction of 1.5%, the detection rate was improved from 64% to 96% at a false-positive rate below 0.0015%. This demonstrates the merits of active learning in practice.

Our paper is structured as follows. Section 2 introduces hypersphere-based anomaly detection and presentes our extension using semi-supervised and active learning strategies. Section 3 reports on empirical results of our approach using real network traffic and attacks. Finally, Section 4 concludes.

2. METHODOLOGY

In this section, we present our methodology. Firstly, we describe how we derive numerical features from network payload data. Then, we review the classical hypersphere-based approach to anomaly detection. We discuss how anomaly detection can be equipped with an active learning strategy to adjust an anomaly threshold. Finally, we propose an integrated method to compute hyperspheres *and* thresholds *simultaneously*.

2.1 From Network Payload to Feature Spaces

The detection of unknown and novel attacks requires an expressive representation of network contents, accessible to means of intrusion detection and machine learning. To this end, we apply a technique for embedding of network payloads in vector spaces derived from concepts of information retrieval [22] and recently applied in the realms of intrusion detection [19]. A network payload $\mathbf{x}$ (the data contained in a network packet or connection) is mapped to a vector space using a set of strings S and an embedding function ϕ. For each string $s \in S$ the function $\phi_s(\mathbf{x})$ returns 1 if s is contained in the payload $\mathbf{x}$ and 0 otherwise. By applying $\phi_s(\mathbf{x})$ for all elements of S we obtain the following map

$$\phi : \mathcal{X} \to \mathbb{R}^{|S|}, \quad \phi : \mathbf{x} \mapsto (\phi_s(\mathbf{x}))_{s \in S}, \tag{1}$$

where $\mathcal{X}$ is the domain of all network payloads. Defining a set S of relevant strings a priori is difficult in advance, as typical patterns of novel attacks are not available prior to their disclosure. As an alternative, we define the set S implicitly and associate S with all strings of length n. The resulting set of strings is often referred to as n-grams.

As a consequence of using n-grams, the network payloads are mapped to a vector space with 256^n dimensions, which apparently contradicts with efficient detection of intrusions. Fortunately, a payload of length T comprises at most $(T - n - 1)$ different n-grams and, consequently, the map ϕ is *sparse*, that is, the vast majority of dimensions is zero. This sparsity can be exploited to derive linear-time algorithms for extraction and comparison of embedded vectors. Instead of operating with full vectors, only non-zero dimensions are considered, where the extracted strings associated with each dimension can be maintained in efficient data structures, such as hash tables [3], Bloom filters [31] or Tries [19].

2.2 Hypersphere-based Anomaly Detection

In this section, we briefly review anomaly detection using one-class support vector machines. In particular we study a variant proposed by Tax and Duin [28] referred to as support vector domain description (SVDD). Several approaches to anomaly detection for network intrusion detection, e.g. [20, 32, 31], can be shown to resemble special cases of the SVDD, given that an appropriate embedding of network events into some vector space as described in the previous section is per-

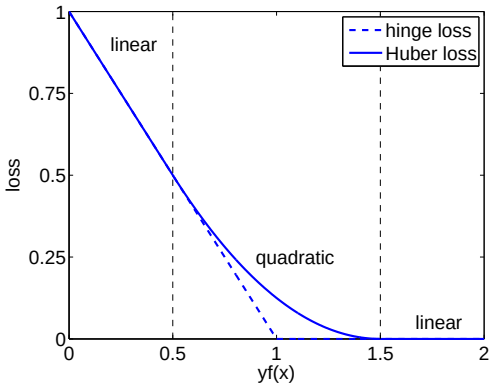

Figure 2: The differentiable Huber loss $\ell_{\Delta=1,\epsilon=0.5}$.

formed. Further discussion and comparison of hypersphere-based anomaly detection is provided in [23, 29].

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 $\mathbf{c}$ and radius R) that contains the normal input data [28], see Figure 1 (left). Given the function

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

the boundary of the hypersphere is described by the set $\{\mathbf{x} : f(\mathbf{x}) = 0 \wedge \mathbf{x} \in \mathcal{X}\}$. That is, the parameters of f are to be chosen such that $f(\mathbf{x}) \leq 0$ for normal data and $f(\mathbf{x}) > 0$ for anomalous points. The center $\mathbf{c}$ and the radius R can be computed accordingly by solving the following optimization problem [28]

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

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 data, while $\eta \to 0$ implies $R \to 0$ and the center $\mathbf{c}$ reduces to the centroid of the data.

Some actions to the SVDD have been proposed to incorporate labeled data in the learning process, e.g. [5, 9, 30, 29], the resulting optimization problems are no longer convex and the proposed optimization in dual space might suffer from duality gaps. Techniques for actively guiding the learning of the SVDD have been not considered so far although active learning for anomaly detection has been studied by [26, 15, 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 [15] aims at detecting rejection categories in the data using as few queries as possible. Finally, the approach taken in [26] combines the former two active learning strategies to find interesting regions in feature space *and* to decrease the error-rate simultaneously.

In this section we devise an efficient strategy to query network events which lie in low-confidence regions of the feature space, hence guiding the security expert in the labeling process. This *active learning* strategy selects an instance of the unlabeled data pool and presents it to the security expert. The selection process is designed to yield the maximal improvement of the actual model.

Our strategy takes unlabeled as well as already labeled examples into account. Without loss of generality, we denote the unlabeled examples by $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and the labeled ones by $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}$, where $n \gg m$. Every labeled example $\mathbf{x}_i$ is annotated with a label $y_i \in \{+1, -1\}$, depending on whether it is classified as benign ($y_i = +1$) or malicous ($y_i = -1$) data.

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 $\mathbf{x}'$ that is closest to the decision hypersphere [1, 33]

$$\begin{aligned} \mathbf{x}' &= \operatorname*{argmin}_{\mathbf{x}_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}} \lambda_1(\mathbf{x}_i) \\ &:= \operatorname*{argmin}_{\mathbf{x}_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}} \frac{|R^2 - \|\phi(\mathbf{x}_i) - \mathbf{c}\|^2|}{\Omega}, \end{aligned} \tag{3}$$

where Ω is a normalization constant and given by $\Omega = \max_i |R^2 - \|\phi(\mathbf{x}_i) - \mathbf{c}\|^2|$.

However, when dealing with many non-stationary outlier and/or attack categories, it is beneficial to identify novel attacks as soon as possible. We translate this into an active learning strategy as follows. Let $A = (a_{st})_{s,t=1,\ldots,n+m}$ be an adjacency matrix, for instance obtained by a k-nearest-neighbor approach, where $a_{ij} = 1$ if $\mathbf{x}_i$ is among the k-nearest neighbors of $\mathbf{x}_j$ and 0 otherwise. Equation (4) implements the above idea and returns the unlabeled instance according to

$$\begin{aligned} \mathbf{x}' &= \operatorname*{argmin}_{\mathbf{x}_t \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}} \lambda_2(\mathbf{x}_t) \\ &:= \operatorname*{argmin}_{\mathbf{x}_t \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}} \frac{\sum_{i=1}^{n} a_{it} + \sum_{j=n+1}^{n+m} y_j a_{jt}}{2k}. \end{aligned} \tag{4}$$

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 (4) 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 (3) and (4) guarantees the active learning to query points of interest. Our final active learning strategy is therefore given by

$$\mathbf{x}' = \operatorname*{argmin}_{\mathbf{x}_t \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}} \tau\lambda_1(\mathbf{x}_t) + (1 - \tau)\lambda_2(\mathbf{x}_t) \tag{5}$$

for $\tau \in [0, 1]$. 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 τ, 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.

2.3 An integrated approach: ActiveSVDD

The active learning strategy from the previous section queries low-confidence points to improve the current hypothesis. The idea is, that the model can be re-trained after quering some points, using the unlabeled *as well as the* the newly labeled data. Unfortunately, the vanilla SVDD cannot make use of labeled data. In this section, we extend the SVDD to support active learning and propose the integrated method ActiveSVDD which determines a hypersphere and a radius simultaneously.

As in Section 2.2, we aim at finding a model $f(\mathbf{x}) = ||\phi(\mathbf{x}) - \mathbf{c}||^2 - R^2$ 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 (2) using both, labeled and unlabeled examples, is given by

$$\begin{aligned} \min_{R,\gamma,\mathbf{c},\xi} \quad & R^2 - \kappa\gamma + \eta_u \sum_{i=1}^{n} \xi_i + \eta_l \sum_{j=n+1}^{n+m} \xi_j \\ \text{s.t.} \quad & \forall_{i=1}^{n} : \|\phi(\mathbf{x}_i) - \mathbf{c}\|^2 \leq R^2 + \xi_i \\ & \forall_{j=n+1}^{n+m} : y_j \left(\|\phi(\mathbf{x}_j) - \mathbf{c}\|^2 - R^2\right) \leq -\gamma + \xi_j \quad (6) \\ & \forall_{i=1}^{n} : \xi_i \geq 0, \\ & \forall_{j=n+1}^{n+m} : \xi_j \geq 0. \end{aligned}$$

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 (6) into an *unconstrained* problem [2, 34] as follows,

$$\begin{aligned} \min_{R,\gamma,\mathbf{c}} \quad & R^2 - \kappa\gamma + \eta_u \sum_{i=1}^{n} \ell\left(R^2 - ||\phi(\mathbf{x}_i) - \mathbf{c}||^2\right) \quad (7) \\ & + \eta_l \sum_{j=n+1}^{n+m} \ell\left(y_j \left(R^2 - ||\phi(\mathbf{x}_j) - \mathbf{c}||^2\right) - \gamma\right). \end{aligned}$$

where $\ell(t) = \max\{-t, 0\}$ is the common hinge loss. Note that the optimization problems in Equations (6) and (7) are equivalent so far. Nevertheless, the non-smoothness of the objective prohibits an efficient optimization. Hence, we 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

$$\ell_{\Delta,\epsilon}(t) = \begin{cases} \Delta - t & : \quad t \leq \Delta - \epsilon \\ \frac{(\Delta+\epsilon-t)^2}{4\epsilon} & : \quad \Delta - \epsilon \leq t \leq \Delta + \epsilon \\ 0 & : \quad \text{otherwise.} \end{cases}$$

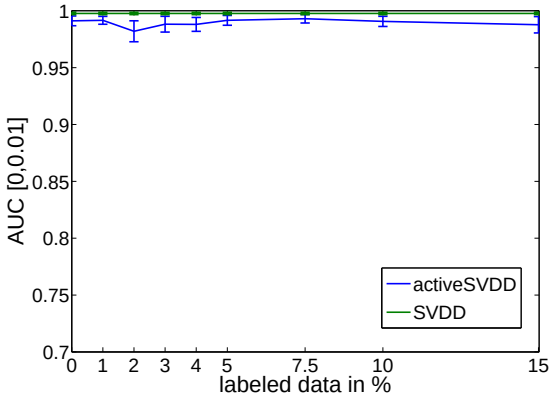

Figure 3: Results for normal vs. malicious.

For our purposes $\Delta = 0$ suffices and the final objective function can be stated as,

$$\begin{aligned} \min_{R,\gamma,\mathbf{c}} \quad & R^2 - \kappa\gamma + \eta_u \sum_{i=1}^{n} \ell_{0,\epsilon}\left(R^2 - ||\phi(\mathbf{x}_i) - \mathbf{c}||^2\right) \quad (8) \\ & + \eta_l \sum_{j=n+1}^{n+m} \ell_{0,\epsilon}\left(y_j \left(R^2 - ||\phi(\mathbf{x}_j) - \mathbf{c}||^2\right) - \gamma\right). \end{aligned}$$

Notice that by rephrasing the problem as an unconstrained, smooth optimization problem, its intrinsic complexity has not changed. However, the local minima of Optimization Problem (8) can now easily be found with gradient-based techniques such as conjugate gradient descent, see Appendix A for details. Note that, in general, unconstrained optimization is also easier to implement than constrained optimization. We will observe the benefit of this approach in the following.

3. EMPIRICAL EVALUATION

We proceed to present an empirical evaluation of our novel method ActiveSVDD for intrusion detection using real network traffic. In particular, we are interested in studying the performance gain attained by our active learning strategy in comparison to the unsupervised formulation of the SVDD [29]. Several approaches to learning-based intrusion detection constitute special cases of the SVDD, e.g. [32, 21, 16], and hence are implicitly reflected in our experiments. The ActiveSVDD is trained by solving Equation (8) using conjugate gradient descent, where the optimization problem underlying the SVDD is solved using SMO [18]. Parameters of the active learning strategy are set to $k = 10$, $\alpha = 0.1$ for simplicity.

3.1 Data Corpus

For our experiments, we consider HTTP traffic recorded within 10 days at Fraunhofer Institute FIRST. The data set comprises 145,069 unmodified connections with an average length of 489 bytes. The incoming byte stream of each connection is mapped to a vector space using 3-grams as detailed in Section 2.1. We refer to the FIRST data as the *normal pool*. The *malicious pool* contains 27 real attack classes generated using the Metasploit framework [12]. It covers 15 buffer overflows, 8 code injections and 4 other at-

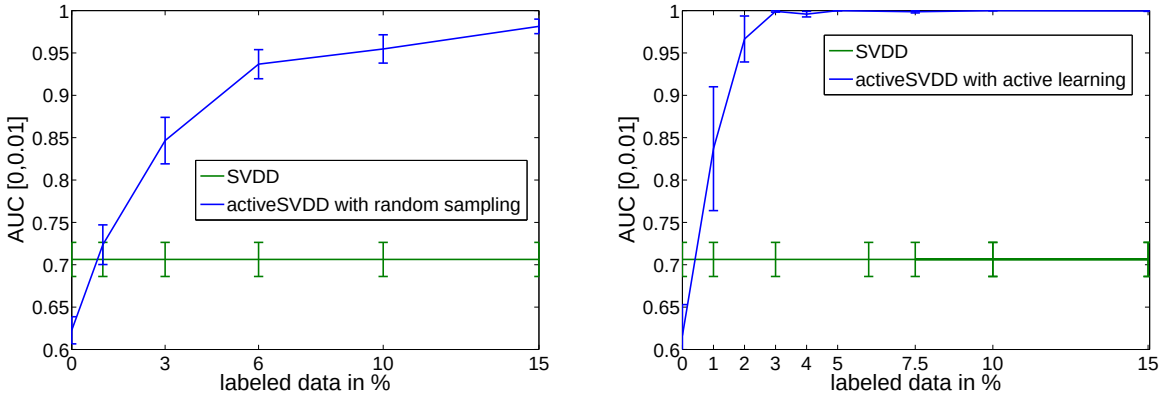

Figure 4: Results for normal vs. cloaked. Left: Random samling. Right: Active learning.

tacks including HTTP tunnels and cross-site scripting. Every attack is recorded in 2–6 different variants using virtual network environments and decoy HTTP servers, where the attack payload is adapted to match characteristics of the normal data pool.

To study the robustness of our approach in a more realistic scenario, we also consider techniques to obfuscate malicious content by adapting attack payloads to mimic benign traffic in feature space [4]. As a consequence, the extracted features deviate less from 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*.

3.2 Active Learning Experiment

In our first experiment we focus on two scenarios: normal vs. malicious and normal vs. cloaked data. For both settings, 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. Note that all attacks in the training data are unknown to the learning methods, unless an active labeling strategy is performed. 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 labeled instances which are selected using random sampling. 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 sufficiently captured by the representation of 3-grams. There is no significant difference between the detectors. 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.

Figure 4 (left) displays the results for normal vs. cloaked data, where network connections to be labeled for the ActiveSVDD are chosen randomly. First of all, the performance of the unsupervised SVDD drops to only 70%, as the cloaked attacks successfully foil the detection process. By contrast, the ActiveSVDD benefits from labeled data and clearly shows a reasonable accuracy. For only 2% labeled data, the ActiveSVDD easily outperforms the vanilla SVDD and for labeling 5% 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 ActiveSVDD is chosen according to the active learning strategy in Equation (5). The unsupervised SVDD does not make use of label information and remains at an $\text{AUC}_{0.01}$ of 70%. Compared to the results for a random labeling strategy (Figure 4, left), the performance of the ActiveSVDD clearly improves for active learning. Using active learning, we need to label only 3% of the data for attaining an almost perfect separation, compared to 30% 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 (5). We compare the number of outliers detected by the combined strategy with the margin-based strategy in Equation (3) (see also [1, 26]) 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 strategy effectively detects malicious traffic much faster than the margin-based strategy.

3.3 Online Application

When employing network intrusion detectors in practice, one faces a steadily increasing amount of unlabeled network data. Holding all instances in memory gets infeasible over time such that means for learning online become a crucial requirement. The next experiment aims at investigating our method ActiveSVDD in an *online learning* scenario, i.e. when the normal data pool steadily increases. To this end,

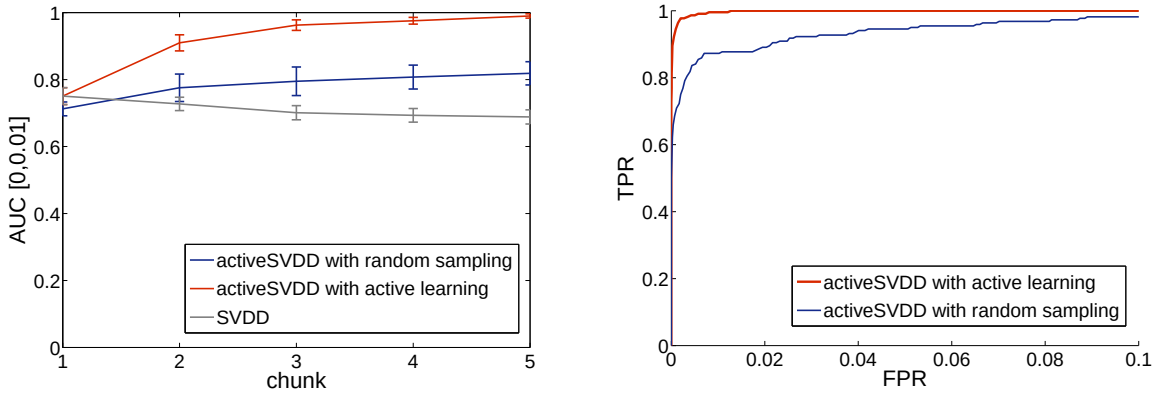

Figure 6: Online application of ActiveSVDD over different chunks. Left: Progress over chunks. Right: ROC curve for all chunks.

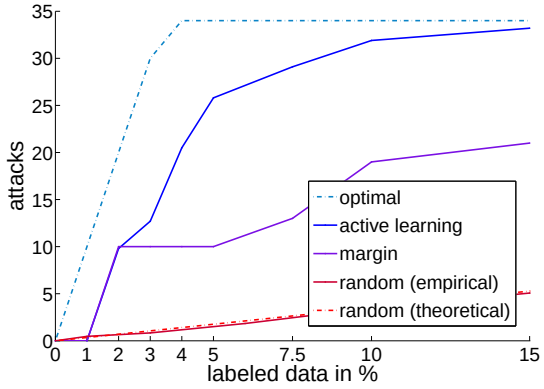

Figure 5: Number of attacks found by different active learning strategies.

we draw a sample of 3,750 network events from the normal pool, where 1,250 connections are used as test set and the remaining data is decomposed into five chunks of equal size for online application. Cloaked attacks are mixed into all samples and we take care that the same attack classes are not present in the training and test data. The ActiveSVDD is then trained on an increasing number of chunks, starting from the first and finally using all five chunks. For each chunk we adjust the active learning strategy such that in average only 10 data points need to be labeled.[1]

Figure 3.2 shows the change in accuracy of the ActiveSVDD over the different chunks. Results are averaged over 10 random draws of data splits. One can see that with increasing amount of network data the active learning strategy steadily drives the learner to high accuracy predictions while the random strategy is too slow to adapt. The vanilla SVDD performs worse since it doesn't profit from the labels. Figure 3.2 shows a ROC curve for the ActiveSVDD and the regular SVDD obtained after learning on all five chunks. By only labeling a fraction of 1.5% the ActiveSVDD enables detecting 96% of the cloaked attacks at a false-positive rate at 0.0015%. By contrast, the vanilla SVDD identifies only 64% attacks at the same false-positive rate.

[1]We feel that such a small amount of data labelings is realistic in the light of massive incoming traffic and the high costs of a human experts.

3.4 Threshold Adaption

The previous experiments demonstrate the advantages of active learning for network intrusion detection. So far, all results have been obtained using our method ActiveSVDD, however, the active learning techniques devised in Section 2 are also applicable for calibrating other learning-based methods. We herein focus on the vanilla SVDD with parameter $\nu = 1$, which corresponds to classical centroid-based anomaly detection, such that results directly transfer to anomaly detectors as Anagram and PAYL.

We again draw a set of 3,750 network connections form the pool of normal data and split the resulting set into a training set of 2,500 connections and a test partition of 1,250 events. Both sets are mixed with cloaked attack instances. The SVDD is then trained on the normal training set. For application of the learned hypersphere to the test set, we evaluate different strategies for determining a radius using random sampling and active learning. In both cases, the selected connections are labeled and a threshold is obtained by computing the mean of all labeled instances.

Figure 7 shows for various levels of labeled data the ROC curve of the SVDD and the computed thresholds that have been derived from the radius outputed by the SVDD. Results have been averaged over 10 random draws of working sets. One can see that even for small amounts of labeled data the active learning strategy finds a reasonable radius while the random strategy and the vanilla SVDD completely fail with false-positive rate of 0.5 and 1 respectively. This result demonstrates that active learning strategies enable calibrating anomaly detection with significantly reduced effort in comparison to random sampling and hence provide a valuable instrument when deploying learning methods in practice.

4. CONCLUSION

In this paper, we proposed to view anomaly detection as an active learning problem to allow for the inclusion of prior

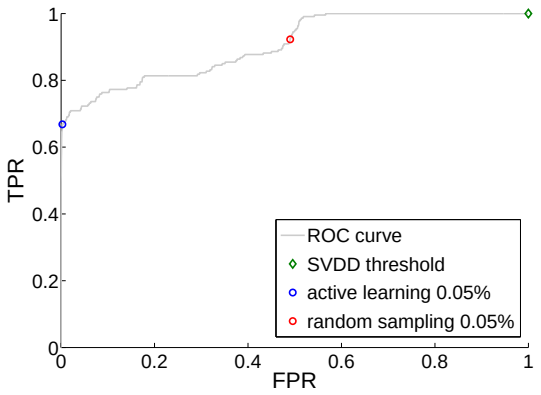

Figure 7: Treshold adaption for the SVDD

and expert knowledge. To reduce the labeling effort for the practitioner, we devised an active learning strategy to query instances that are not only close to the boundary of the hypersphere but also likely members of novel rejection categories. To use labeled as well as unlabeled instances in the training process, we proposed ActiveSVDDs as a generalization of SVDDs. The resulting unconstraint, smooth optimization problem can be optimized with efficient gradient-based techniques.

Empirically, we showed for network intrusion detection, that rephrasing the unsupervised problem setting as an active learning task is worth the effort. ActiveSVDDs 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 ActiveSVDD and spares practitioners from labeling unnecessarily many data points. For experiments on sequentially arriving data chunks, the ActiveSVDDs achieve a perfect separation of normal and attack data and outperform its unsupervised counterpart significantly.

Acknowledgements

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.

5. REFERENCES

[1] M. Almgren and E. Jonsson. Using active learning in intrusion detection. *Proc. IEEE Computer Security Foundation Workshop*, 2004.

[2] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the International Workshop on AI and Statistics*, 2005.

[3] M. Damashek. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843–848, 1995.

[4] P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee. Polymorphic blending attacks. In *Proceedings of USENIX Security Symposium*, 2006.

[5] C.-H. Hoi, C.-H. Chan, K. Huang, M. Lyu, and I. King. Support vector machines for class representation and discrimination. In *Proceedings of the International Joint Conference on Neural Networks*, 2003.

[6] K. L. Ingham, A. Somayaji, J. Burge, and S. Forrest. Learning DFA representations of HTTP for protecting web applications. *Computer Networks*, 51(5):1239–1255, 2007.

[7] M. Kloft and P. Laskov. A poisoning attack against online anomaly detection. In *NIPS Workshop on Machine Learning in Adversarial Environments for Computer Security*, 2007.

[8] C. Kruegel, G. Vigna, and W. Robertson. A multi-model approach to the detection of web-based attacks. *Computer Networks*, 48(5), 2005.

[9] 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.

[10] M. Mahoney. Network traffic anomaly detection based on packet bytes. In *Proc. of ACM Symposium on Applied Computing*, pages 346 – 350, 2003.

[11] M. Mahoney and P. Chan. PHAD: Packet header anomaly detection for identifying hostile network traffic. Technical Report CS-2001-2, Florida Institute of Technology, 2001.

[12] K. Maynor, K. Mookhey, J. F. R. Cervini, and K. Beaver. Metasploit toolkit. In *Syngress*, 2007.

[13] Microsoft. Microsoft security intelligence report: January to June 2008. Microsoft Corporation, 2008.

[14] D. Moore, C. Shannon, and J. Brown. Code-Red: a case study on the spread and victims of an internet worm. In *Proc. of Internet Measurement Workshop (IMW)*, pages 273–284, 2002.

[15] D. Pelleg and A. Moore. Active learning for anomaly and rare-category detection. *Proc. Advances in Neural Information Processing Systems*, 2004.

[16] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee. McPAD: A multiple classifier system for accurate payload-based anomaly detection. *Computer Networks*, 2009. in press.

[17] R. Perdisci, D. Dagon, W. Lee, P. Fogla, and M. Sharif. Misleading worm signature generators using deliberate noise injection. In *Proc. of IEEE Symposium on Security and Privacy*, pages 17–31, 2006.

[18] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: support vector learning*, 1999.

[19] K. Rieck and P. 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] K. Rieck and P. Laskov. Language models for detection of unknown attacks in network traffic. *Journal in Computer Virology*, 2(4):243–256, 2007.

[21] K. Rieck, S. Wahl, P. Laskov, P. Domschitz, and K.-R.

Müller. A self-learning system for detection of anomalous sip messages. In *Principles, Systems and Applications of IP Telecommunications (IPTCOMM), Second International Conference*, LNCS, pages 90–106, 2008.

[22] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[23] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[24] C. Shannon and D. Moore. The spread of the Witty worm. *IEEE Security and Privacy*, 2(4):46–50, 2004.

[25] Y. Song, M. Locasto, A. Stavrou, A. Keromytis, and S. Stolfo. On the infeasibility of modeling polymorphic shellcode. In *Conference on Computer and Communications Security (CCS)*, pages 541–551, 2007.

[26] J. W. Stokes and J. C. Platt. Aladin: Active learning of anomalies to detect intrusion. Technical report, Microsoft Research, 2008.

[27] Symantec. Symantex report on the underground economy: July 07 to June 08. Symantec Corporation, 2008.

[28] D. M. Tax and R. P. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004.

[29] D. M. Tax. *One-class classification*. PhD thesis, Technical University Delft, 2001.

[30] J. Wang, P. Neskovic, and L. N. Cooper. Pattern classification via single spheres. *Computer Science: Discovery Science (DS)*, 2005.

[31] K. Wang, J. Parekh, and S. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Adances in Intrusion Detection (RAID)*, pages 226–248, 2006.

[32] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In *Recent Adances in Intrusion Detection (RAID)*, pages 203–222, 2004.

[33] 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.

[34] 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

A. GRADIENT COMPUTATION

In this section, we compute the gradient of ActiveSVDD. For problem (6), the slacks can be expressed as

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

respectively. Furthermore the derivative of the Huber loss $\ell_{\Delta,\epsilon}$ is given by

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

For notational convenience, we focus on the Huber loss for $\ell_{\Delta=0,\epsilon}(t)$. 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 $\mathbf{c}$ yields

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

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

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

Substituting the partial gradients, we resolve the gradient of Equation (7) with respect to the primal variables:

$$\frac{\partial EQ7}{\partial R} = 2R + \eta_u \sum_{i=1}^{n} \frac{\partial \xi_i}{\partial R} + \eta_l \sum_{j=n+1}^{n+m} \frac{\partial \xi_j}{\partial R}, \tag{9}$$

$$\frac{\partial EQ7}{\partial \gamma} = -\kappa + \eta_l \sum_{j=n+1}^{n+m} \frac{\partial \xi_j}{\partial \gamma}, \tag{10}$$

$$\frac{\partial EQ7}{\partial \mathbf{c}} = \eta_u \sum_{i=1}^{n} \frac{\partial \xi_i}{\partial \mathbf{c}} + \eta_l \sum_{j=n+1}^{n+m} \frac{\partial \xi_j}{\partial \mathbf{c}}. \tag{11}$$

The above equations can be plugged directly into off-the-shelf gradient-based optimization tools to optimize Equation (7) in the input space for the identity $\phi(\mathbf{x}) = \mathbf{x}$. 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 non-linear feature embeddings. An application of the representer theorem shows that the center $\mathbf{c}$ can be expanded as

$$\mathbf{c} = \sum_i \alpha_i \phi(\mathbf{x}_i) + \sum_j \alpha_j y_j \phi(\mathbf{x}_j). \tag{12}$$

According to the chain rule, the gradient of Equation (7) with respect to the $\alpha_{i/j}$ is given by

$$\frac{\partial EQ7}{\partial \alpha_{i/j}} = \frac{\partial EQ7}{\partial \mathbf{c}} \frac{\partial \mathbf{c}}{\partial \alpha_{i/j}}.$$

Using Equation (12), the partial derivatives $\frac{\partial \mathbf{c}}{\partial \alpha_{i/j}}$ resolve to

$$\frac{\partial \mathbf{c}}{\partial \alpha_i} = \phi(\mathbf{x}_i) \quad \text{and} \quad \frac{\partial \mathbf{c}}{\partial \alpha_j} = y_j \phi(\mathbf{x}_j), \tag{13}$$

respectively. Applying the chain-rule to Equations (9),(10),(11), and (13) gives the gradients of Equation (7) with respect to the $\alpha_{i/j}$.