Reducing False Alarms in Cyber Attack Detection by Using Support Vector Data Description

Pankaj R. Telang
Deovrat Kakde
SAS Institute Inc.
100 SAS Campus Drive, Cary, NC 27513, USA
{pankaj.telang, dev.kakde}@sas.com

ABSTRACT

Continual rise in cyber attacks against enterprises indicates that traditional signature-based approaches for attack detection are insufficient. It is important to develop effective signature-less data-mining approaches for detecting attacks in enterprise networks. We apply Support Vector Data Description (SVDD), a single-class classification technique, for attack detection. On an enterprise network data set, we show that a single SVDD model can lead to a large number of false alarms when the training data set contains multiple disjoint clusters. To address this issue, we propose a novel attack detection approach that combines clustering with SVDD for attack detection. We demonstrate the problem and the effectiveness of our approach in reducing the false alarms on a toy data set. Further, we present an extensive evaluation of our approach on a real-world enterprise network data set. Our approach includes an efficient way of tuning the SVDD model hyperparameter and uses a fast training algorithm. This enables our method to be practically usable in the enterprise settings that demand fast processing speed and automation.

Keywords

Cybersecurity, SVDD, anomaly detection

1. INTRODUCTION

Cyber attacks on enterprises (*businesses and government agencies*) continue to rise. The attackers aggressively find new vulnerabilities in the devices that run on an enterprise network and exploit them as zero-day attacks. Traditional signature-based approaches for attack detection are ineffective against such newly crafted attacks. In contrast, data mining methods for cyber attack detection do not employ attack signatures, and can effectively detect novel attacks.

The primary intent of cyber attack detection is to classify activities as either *normal or attack*. In a typical enterprise, data about normal network activity is abundant, where as attack data does not exist or is extremely limited. Such data, which is dominant with observations that correspond to a single class such as normal network activity in this case, is called as the single-class data. Support vector data description [21] is an established single-class classification technique. In this paper, we use SVDD for cyber attack detection.

An important issue with the data-mining oriented systems in cybersecurity is the large number of false alarms (false positives), that is, a large number of normal activities misclassified as attack activities. Since each attack is typically investigated by a human analyst, a large number of false alarms result in high labor costs and wasted human effort. Further, it reduces the analysts' trust in the system.

We show that a single SVDD model comprising all of the enterprise network activities leads to a large number of false positives. A typical enterprise network contains a variety of devices, such as servers, laptops, desktops, smart phones, and security cameras. These devices exhibit widely different network behaviors. We posit that the large number of false positives in a single SVDD model is due to the large difference in the enterprise device behaviors. To address this issue, we propose a novel approach that employs unsupervised clustering with several SVDD models, corresponding to each cluster for detecting attacks. In comparison to using a single SVDD model, our approach significantly reduces the number of false positives. We evaluate our approach on a real-world network data set.

Contributions: We propose an approach that combines unsupervised clustering and SVDD classification for extracting cyber attacks from enterprise network activities. We present an extensive evaluation of the proposed approach on a publicly available enterprise network data set. The evaluation shows that our approach leads to a significant reduction in the false positive rate.

Organization: Section 2 introduces the SVDD technique followed by the mathematical formulation and guidelines for parameter selection in Section 2.1. Section 3 demonstrates the problem of large number of false alarms on a toy dataset and introduces our proposed approach. Section 4 describes our proposed approach that combines unsupervised clustering with SVDD. Section 5 presents an extensive evaluation of our approach using a real-world network data set. Section 6 presents the related work. Finally, Section 7 discusses the conclusions and the future work.

2. BACKGROUND

Support vector data description is a machine learning tech-

KDD 2019 Workshop on Learning and Mining for Cybersecurity (LEMINCS'19) August 5th, 2019, Anchorage, AK

ACM ISBN .

DOI:

nique that is used for single-class classification and anomaly detection. First introduced by Tax and Duin [21], SVDD's mathematical formulation is equivalent to one-class support vector machines (OCSVM), which is the one-class variant of support vector machines using Gaussian kernel [19]. The use of SVDD is popular in domains where the majority of data belongs to a single class, it is multivariate and it is not possible to make any distributional assumptions about data, such as the multivariate normal. For example, SVDD is useful for analyzing sensor readings from reliable equipment where almost all the readings describe the equipment's normal state of operation.

In its simplest form, SVDD builds a minimum-radius hypersphere around the one-class training data. For a new observation, its distance from the center of the hypersphere is computed. If it is more than the radius, the observation is designated as an outlier. Otherwise, the observation is an inlier. A more flexible data description can be obtained using kernel functions. Such description is not circular and boundary of such data description closely follows the geometry of the data.

Several researchers have proposed using SVDD for multivariate process control [20, 1]. Other applications of SVDD involve machine condition monitoring [24, 26] and image classification [18].

We use a data set with the normal network activity for developing a SVDD model. An outlier detected by this SVDD model is a potential cyber attack.

2.1 Mathematical Formulation of SVDD

This section describes the mathematical formulation of the SVDD model, how to apply the SVDD model to score new observations and the guidelines for setting various parameters. The mathematical formulation follows Tax and Duin [21] and Kakde et al. [12].

2.1.1 Primal Formulation

As outlined in Section 2, the SVDD model in its simplest form builds a minimum radius hypersphere around the training data. The SVDD model can be expressed as an optimization problem. Its objective function is:

$$\min R^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$

In Equation 1, R is a *radius* and represents the decision variable, $C = \frac{1}{nf}$ is a penalty constant where n is the number of observations in the training data and f is a fraction of expected outliers. The penalty constant controls the tradeoff between the volume and the errors. The constraints of the optimization problem are:

$$\|x_i - a\|^2 \leq R^2 + \xi_i, \forall i = 1, \ldots, n \tag{2}$$

$$\xi_i \geq 0, \forall i = 1, \ldots, n \tag{3}$$

In Equations 2 and 3, $x_i \in \mathbb{R}^m$ where m is the number of variables, $i = 1, \ldots, n$ are the observations from the training data, ξ_i is a slack for each observation and a is the center of the hypersphere.

2.1.2 Dual Formulation

The dual formulation of the above optimization problem employs Lagrange multipliers. The dual objective function is:

$$\max \sum_{i=1}^{n}\alpha_i(x_i \cdot x_i) - \sum_{i,j}\alpha_i\alpha_j(x_i \cdot x_j) \tag{4}$$

In Equation 4, $\alpha_i \in \mathbb{R}$ are the Lagrange constants. The constraints of the dual optimization problem are:

$$\sum_{i=1}^{n}\alpha_i = 1 \tag{5}$$

$$0 \leq \alpha_i \leq C, \forall i = 1, \ldots, n \tag{6}$$

The solution to the above optimization problem consists of values α_i for all observations $i = 1 \ldots n$. An observation can be designated as an inlier or an outlier based on the value of α_i as shown in Table 1.

Table 1: SVDD Scoring Criteria

Criteria	Observation Position
$\sum_{i=1}^{n}\alpha_i x_i = a$	Center
$\|x_i - a\| < R \rightarrow \alpha_i = 0$	Inside
$\|x_i - a\| = R \rightarrow 0 < \alpha_i < C$	Boundary
$\|x_i - a\| > R \rightarrow \alpha_i = C$	Outside

From Table 1, observe that the center a only depends upon the observations with $\alpha_i > 0$. Such observations are called *support vectors*. We denote the set of support vectors as Ψ.

The above formulation of SVDD computes a circular data boundary around the training data which can potentially include a significant amount of space with sparsely distributed training observations. A model with such a boundary leads to a large number of false positives. Thus, instead of circular, a compact bounded outline around the training data is desirable. A kernel function enables calculation of such a compact outline.

2.1.3 Flexible Data Description

The SVDD model is made flexible by replacing the inner product $(x_i \cdot x_j)$ with a suitable kernel function $K(x_i, x_j)$. The boundary of such data description closely follows the geometry of the training data set. We employ the Gaussian kernel function from Equation 7.

$$K(x_i, x_j) = \exp\frac{-\|x_i - x_j\|^2}{2s^2} \tag{7}$$

In Equation 7, s is the Gaussian bandwidth parameter.

When using a kernel function, a threshold R^2 value is calculated to score the observations. Equation 8 calculates the threshold R^2.

$$R^2 = K(x_k, x_k) - 2\sum_{i}\alpha_i K(x_i, x_k) + \sum_{i,j}\alpha_i\alpha_j K(x_i, x_j) \tag{8}$$

In Equation 8, x_k is a support vector such that $x_k \in \Psi_{<C}$, where $\Psi_{<C} \subset \Psi$ such that $\alpha_k < C$.

2.1.4 Scoring

In order to score the observations, for each observation z in the scoring data set, Equation 9 computes the distance value $\delta^2(z)$.

$$\delta^2(z) = K(z,z) - 2\sum_i \alpha_i K(x_i,z) + \sum_{i,j} \alpha_i \alpha_j K(x_i,x_j) \quad (9)$$

Observations in the scoring data set with $\delta^2(z) > R^2$ are outliers.

2.2 Guidelines for Parameter Selection

Note that the SVDD formulation with Gaussian kernel function has two parameters: the outlier fraction f and the Gaussian bandwidth parameter s. We describe the guidelines for selecting an appropriate value for the bandwidth and outlier fraction parameters.

2.2.1 Gaussian Bandwidth Parameter s

If the value of the outlier fraction f is kept constant, the number of support vectors that the SVDD algorithm identifies is a function of the Gaussian bandwidth parameter s. At a very low value of s, the number of support vectors is very high, approaching the number of observations. As the value of s increases, the number of support vectors decreases. At lower values of s, the data boundary is extremely wiggly. As s is increased, the data boundary becomes less wiggly, and it starts to follow the shape of the data. At higher values of s, the data boundary starts becoming spherical.

In support vector machines, cross validation is a widely used technique [9] for selecting the Gaussian bandwidth parameter. It requires a labeled training data set that contains both normal and outlier classes. If such a data set is not available, then cross validation is infeasible. In cybersecurity, labeled data sets are rare. Even when a labeled data set is available, it generally contains a large number of normal (non-attack) class observations, but very few or no outlier (attack) class observations. This makes cross validation infeasible.

Unlike cross-validation, unsupervised methods for kernel bandwidth selection exist that do not require labeled data. Kakde et al. [12] present the peak criterion method for bandwidth selection and demonstrate its superior performance as compared to other unsupervised methods, such as coefficient of variation [6], maximum distance [14] and the distance to the farthest neighbor [25]. However, peak criterion method requires a grid search of bandwidth values, which is computationally expensive and also sensitive to the starting values in the grid search.

Chaudhuri et al. [3] present the mean criterion method for bandwidth. For a training data set with n observations, the mean criterion method uses Equation 10 to compute the bandwidth value.

$$s = \sqrt{\frac{\bar{D}^2}{\ln \frac{n-1}{\delta^2}}} \quad (10)$$

$\bar{D}^2$ is computed using Equation 11.

$$\bar{D}^2 = \frac{\sum_{i<j} ||x_i - x_j||^2}{\binom{n}{2}} \quad (11)$$

In Equation 11, x_i and x_j represent any two observations in the training data set and δ is the tolerance parameter.

Liao et al. [16] describe a modified mean criterion method, which is an extension of the mean criterion method. The modified mean criterion method computes the tolerance parameter δ of the mean criterion method using Equation 12.

$$\delta = -0.14818008\phi^4 + 0.284623624\phi^3 - 0.252853808\phi^2 + 0.159059498\phi - 0.001381145 \quad (12)$$

In Equation 12, $\phi = \frac{1}{\ln(n-1)}$ and n is the number of observations in the training data set. An SVDD model trained with a modified mean criterion bandwidth is reasonably accurate for many data sets. Although it can be less accurate than a model that is trained with the peak criterion bandwidth, it provides a fast way of computing the bandwidth value, which is essential in automating any analytical work flow. We use the modified mean criterion method on our evaluation of the SVDD for detecting cyber attacks.

2.2.2 Fraction Outlier f

The value of the outlier fraction f inversely relates to the penalty constant C. Per Equations 5 and 6, a higher value of f, which translates to a lower value of C, increases the constraint on α_i, and leads to a larger number of support vectors. Recall that a lower C value decreases the volume that is enclosed by the SVDD boundary.

A data description obtained using a value of f that is significantly different from its true value can lead to incorrect results. A value of f that is more than its true value leads to a higher misclassification rate because observations are incorrectly classified as outliers. It also leads to a smaller value of threshold R^2. On the other hand, a value of f that is less than its true value also leads to a higher misclassification rate because real outliers are classified as inliers.

In this paper we assume that the training data set that is used for building the SVDD model has no outliers and all observations correspond to the normal network activity. Hence we set the value of outlier fraction f to a very low value of 1e−6.

3. ANALYSIS ON TOY DATA SET

This section analyzes the problem of the large number of false positives that occur when applying SVDD, and outlines our proposed approach.

3.1 Toy data set

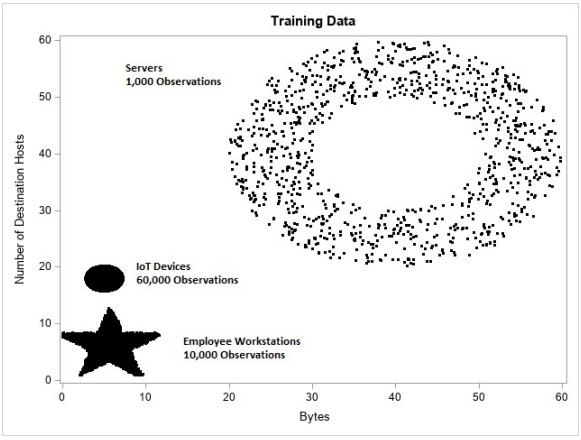

Figure 1: Training Data

Figure 1 shows a toy data set that we simulated to mimic the network activities in an enterprise. It contains two variables: bytes (representing the amount of data sent by a device); and hosts (representing the number of distinct destination hosts with which a device communicates). The toy data contains three disjoint clusters, which represent three different kinds of devices: enterprise servers, employee workstations, and IoT devices. The clusters contain different number of devices: 1,000 servers, 10,000 workstations, and 60,000 IoT devices. The proportion of the device counts is representative of a typical enterprise, and the variable (bytes and hosts) values in different clusters is representative of the network activities of the corresponding device types. For example, end-user workstations typically have fewer bytes and hosts compared to the enterprise servers. We intentionally use the star, oval and a donut shape in the toy data set to highlight the versatility of the SVDD algorithm to address differing geometry of constituting clusters in the training data set. As we see in Section 3.2 and Section 3.3, the SVDD model is able to capture the shape of different clusters in both the single-model approach and the multi-model approach.

3.2 Single SVDD Model Approach

We train a single SVDD model on the toy data set. The model uses a Gaussian bandwidth value that is computed using the modified mean criterion method and the outlier fraction value of 1e−6. To evaluate the model quality, we score a 200 × 200 data grid using the model. Figure 2 shows the scatter plot of the scoring results. The gray area indicates the observations that are scored as outliers, and the black area indicates observations that are scored as inliers.

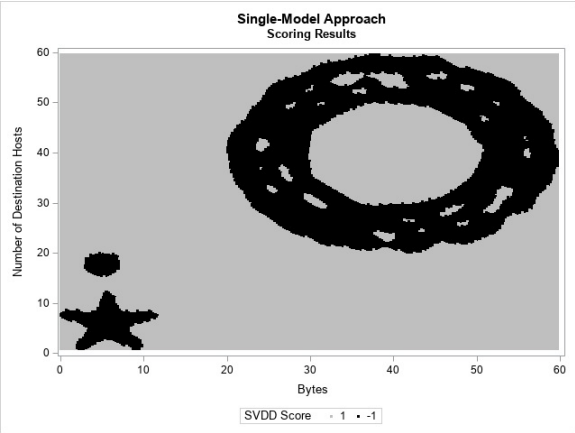

Figure 2: Single-Model Approach: Scoring Results

A comparison between the training data from Figure 1 and the scoring results from Figure 2 indicates that the single SVDD model correctly classifies observations that are outside the training data as outliers. However, notice the gray cavities in the Servers cluster. An observation that is part of any of these cavities is misclassified as an outlier. The observations from all such gray cavities contribute to the total false positives. Also, note that the gray cavities are only present in the Servers cluster but not in the other two clusters. The Servers cluster is the largest among the three clusters and contains the least number of observations. This indicates that the false positive rate can be reduced if one can obtain a data description that is devoid of cavities.

Reduction in cavities can be a challenge if a single bandwidth value is used to train an SVDD model for the entire data set, which has multiple clusters of varying scale and densities. Hence, in the next approach, we train multiple SVDD models, one for each cluster in the training data.

3.3 The Multiple SVDD Model Approach

Instead of training a single SVDD model, in this section we train multiple SVDD models.

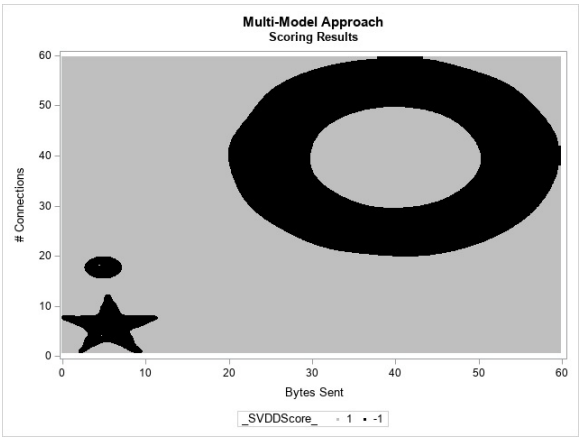

Figure 3: Multi-Model Approach: Scoring Results

We employ unsupervised clustering to cluster the toy data set into three clusters. We now train three SVDD models– one model per cluster. Similar to the single SVDD model, these models use Gaussian bandwidth values that are computed using the modified mean criterion method and the outlier fraction value of 1e−6. We consider an observation as outlier if it is scored as an outlier by all of the three SVDD models. We score a 200 × 200 data grid using each of the model. Figure 3 shows the scatter plot of the scoring results. In comparison to the single SVDD model results from Figure 2, the plot contains significantly fewer and smaller gray holes, that is, significantly fewer false positive.

As outlined in Section 2.2.1 bandwidth value computations using the modified mean criterion method are based on distances between observations in the training data. Hence when data consists of multiple clusters of varying scale and density, the bandwidth computation involves distances between observations within and across clusters. Hence data description obtained using such bandwidth value, may not accurately capture the geometry of each individual cluster. Where as, in case of the multi-model approach, bandwidth for each cluster is based solely on the distances between observations which are within the specific cluster. Hence data description obtained using such bandwidth values is more accurate and it closely follows the boundary of individual clusters. Note bandwidth values used in single SVDD model versus the bandwidth values used in the multiple SVDD model. In single SVDD model, the bandwidth value of 2.19864 is smaller than the bandwidth value of 5.75628 used for the server cluster in case of multiple SVDD model approach. This explains why the description of workstation cluster in multi model approach is devoid of any cavities. Similarly, if one compares the bandwidth values of 0.50044 used for the IoT device cluster and bandwidth value of 1.0458 used for

employee workstation cluster used in the multi model approach, against the single model bandwidth of 2.19864, the lower bandwidth values explain the sharper boundaries for these two clusters in Figure 3.

4. PROPOSED APPROACH

This section builds on the findings from Section 3 and presents our proposed approach in detail.

Although an SVDD model using a kernel function computes a compact boundary around the training data, if the data inherently contains multiple disjoint clusters of significantly different density and scale, a single model constructed with one bandwidth value might lead to false positives in the sparse and/or larger cluster region.

To address this issue, we propose an approach that employs clustering prior to using SVDD for outlier detection. Figure 4 shows our proposed approach. In this approach, a separate SVDD model is fitted for each cluster. A bandwidth value that is computed for each cluster using the modified mean criterion takes the scale of the data in that cluster into account. Hence, it is expected to provide a better, more compact description, with fewer cavities. Next, we describe the steps of our approach.

- First, our approach standardizes the training data set. For each feature, we compute the mean and standard deviation. The standardized value is computed by subtracting the mean from the feature value and dividing the result by the standard deviation. This effectively shifts each feature's mean to zero and standard deviation to one. The standardization is necessary if the features have significantly different scales.
- Second, our approach clusters the training data set using an unsupervised clustering algorithm such as KMeans or DBScan. If the number of clusters in the data set is unknown, we compute the optimal number of clusters using existing approaches such as ABC criterion [22].
- Third, our approach trains one SVDD model per training data set cluster. The bandwidth for the SVDD model is computed using an existing methods such as mean criterion or the modified mean.
- Fourth, for scoring new activities, our approach first standardizes the scoring data set. The standardization uses means and standard deviations of the features that are computed during the first step. The approach then scores the data set using the SVDD model of each cluster. An observation is considered an outlier if it is scored as an outlier by all of the SVDD models.

5. EVALUATION

This section presents an extensive evaluation of our proposed approach on a publicly available network data set.

5.1 Data Set

Our evaluation uses a network data set named CTU-13 [7] that was captured at Czech Technical University. The data set contains a mixture of normal network traffic and botnet traffic, and it is labeled. It includes traffic captured from running various botnets such as Neris, Rbot, Virut, and Menti. The traffic data is in the NetFlow format [10]. We now describe the variables in the data set.

Start time: the time at which the first packet is sent in the NetFlow

Duration: the time between the first and the last packet in the NetFlow

Protocol: the Internet Protocol (IP) such as TCP, UDP, ICMP and ARP in the NetFlow

Source and destination address: the IP addresses of the source and destination devices in the NetFlow

Source and destination port: the ports used by the source and destination devices in the NetFlow

Direction: the direction of network traffic in the NetFlow: unidirectional (from source to destination or from destination to source); or bidirectional

State: the set of TCP flags such SYN, ACK, FIN, etc. of the sender and the receiver in the NetFlow

Source and destination type of service: the type of service on the packets that are sent by the source and destination devices in the NetFlow

Total packets: the total number of packets in the NetFlow

Total bytes: the total number of bytes that are sent by both the source and destination devices in the NetFlow

Source bytes: the bytes sent only by the source device in the NetFlow

Label: the ground-truth type (normal, background, or botnet-related) of the NetFlow

In this paper, we refer to all NetFlows labeled as normal or background as *normal flows* or *non-attack flows* and those labeled as botnet-related as *botflows* or *attack flows*. We focus the evaluation on the TCP, UDP and ICMP protocol flows because all of the botnets use one of these protocols. Table 2 shows the counts of the normal flows and botflows in the CTU-13 data set that use the TCP, UDP, or ICMP protocols.

Table 2: Flow Counts in the CTU-13 data set

Type	Count	Percent
Normal Flows	19,455,505	98%
Botnet Flows	444,699	2%
Total Flows	19,900,204	

5.2 Features

NetFlow aggregates network traffic between a pair of source and destination devices. However, a typical cyber attack involves network activity between more than one pair of devices. For example, consider a device that is infected with malware that performs a host scan on a subnet. Such activity creates multiple NetFlows with the source IP and port of the infected device and destination IPs and ports of the scanned hosts. Therefore, instead of classifying an individual NetFlow that is sent by a source device as attack or non-attack, we aggregate the NetFlows that originate from each source over a time window and classify the aggregated traffic as normal activity or attack activity. The evaluation employs a fixed hour-of-day time window for aggregating the NetFlows that originate from each source IP address. We

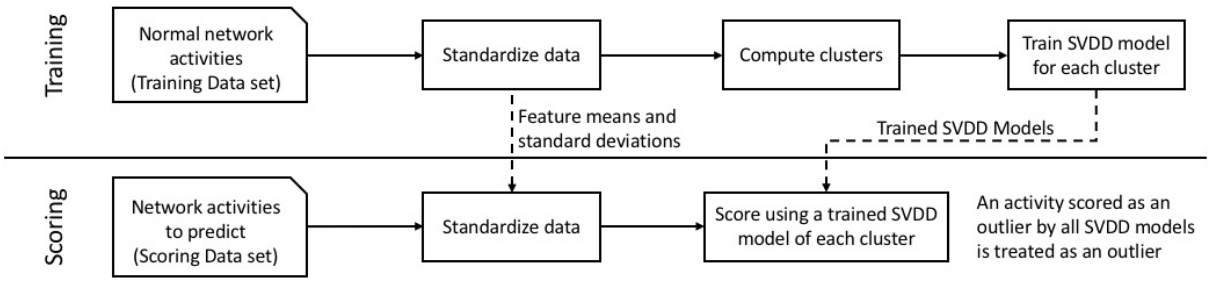

Figure 4: Approach

leave the study of the impact of different time windows on the model accuracy as a future work.

Next, we describe the features we extract from the aggregated data.

Destination hosts: the number of distinct destination device IP addresses. A high value of destination hosts is indicative of the source device performing a host scan.

Maximum destination ports: the maximum number of ports on any of the destination IP addresses. A high value of maximum destination ports is indicative of the source device performing a port scan.

DNS events: the number of flows with DNS destination port (53). A high value of DNS events is indicative of the source device attempting to use the DNS protocol for data exfiltration.

SSH scanning: the number of distinct destination IPs with destination port SSH (22). A high value of SSH scanning is indicative of the source device performing a scan to locate SSH servers.

Telnet scanning: the number of distinct destination IPs with destination port Telnet (23). A high value of Telnet scanning is indicative of the source device performing a scan to locate Telnet servers.

FTP scanning: the number of distinct destination IPs with destination port FTP (21). A high value of FTP scanning is indicative of the source device performing a scan to locate FTP servers.

Database scanning: the number of distinct destination IPs with a destination port of typical databases such as MySQL (3306) or Oracle (1521). A high value of database scanning is indicative of the source device performing a scan to locate database servers.

Application server scanning: the number of distinct destination IPs with a destination port of typical web application servers (80, 443). A high value of application server scanning is indicative of the source device performing a scan to locate application servers.

Domain controller scanning: the number of distinct destination IPs with a domain controller destination port. A high value of domain controller scanning is indicative of the source device performing a scan to locate domain controllers.

Domain controller events: the number of flows with a domain controller destination port. A high value of domain controller events is indicative of the source device attempting to access and extract data from domain controllers.

ICMP scanning: the number of destination IPs in which the protocol is ICMP. A high value of ICMP scanning is indicative of the source device performing a scan to locate hosts that are up.

UDP packets: the total number of packets in which the protocol is UDP. A high value of UDP packets is indicative of the source device performing anomalous activity using the UDP protocol.

Bytes: the total number of bytes that are sent and received. A high value of bytes is indicative of the source device attempting to exfiltrate data.

Flow duration: the sum, mean and standard deviation of the flow durations. The attack flows can have different sum, mean and standard deviations of the flow durations as compared to the normal flows.

Sender states: several features one per distinct sender connection state in terms of the TCP flags set in the NetFlow. The TCP flags include: synchronize (S), acknowledge (A), finish (F), urgent (U), push (P), and reset (R) [5]. The feature value is the count of flows with the given sender state value. For example, sender_FSPA feature is the count of flows in which the sender state is FSPA. The attack flows can have different sender states as compared to the normal flows.

Receiver states: several features one per distinct receiver connection state. The feature value is the count of flows having the given receiver state value. For example, receiver_SRA feature is the count of flows where the receiver state is SRA. The attack flows can have different receiver states as compared to the normal flow.

In addition to the above features, we compute the total number of botflows for each source device and each hour of day. The presence of a botflow implies that the source device is infected by malware and is being used for a cyber attack. We refer to the data set with the features as CTU_FEATURES. The CTU_FEATURES data set contains an observation for each source device and each hour of the day. It has 2,031,488 source devices and NetFlows that span 10 days.

Note that we only present a core set of features, which is not exhaustive. Our set of features can guide the design of additional features. For example, some Cyber attacks employ DNS protocol for for data exfiltration. A feature that computes the sum of bytes sent by a device over the DNS protocol can be effective in detecting such an attack.

5.3 The Single-SVDD Model

This section describes the details of the single-SVDD model and its results.

To create a training data set, we split CTU_FEATURES into two data sets: CTU_BOT and CTU_NORM. CTU_BOT contains those observations for which the number of botflows is greater than zero, and CTU_NORM contains the rest of the observations which are normal. CTU_NORM is further split into two data sets: CTU_NORM_TRAIN and CTU_TEST, containing 80% and 20% of the observations, respectively. We then combine CTU_BOT and CTU_NORM_TRAIN to create CTU_TRAIN data set.

Table 3: Performance Results of the Single-SVDD Model

Botflows	TP	FP	TN	FN	TPR	FPR	FP/TP
1	164	36110	887518	39	0.808	0.039	220.18
2	164	36110	887531	26	0.863	0.039	220.18
3	164	36110	887537	20	0.891	0.039	220.18
4	163	36111	887538	19	0.896	0.039	221.54
5	163	36111	887554	3	0.982	0.039	221.54
6	156	36118	887555	2	0.987	0.039	231.53
7	156	36118	887555	2	0.987	0.039	231.53
8	146	36128	887555	2	0.986	0.039	247.45
9	146	36128	887557	0	1.0	0.039	247.45
10	146	36128	887557	0	1.0	0.039	247.45

We employ the SVDD procedure that is available in the SAS® Visual Data Mining and Machine Learning 8.3 [11]. We use the modified mean criterion method for bandwidth calculation, and the stochastic subset solver [2] that the SAS SVDD procedure provides. Stochastic subset solver is an iterative method for SVDD training that uses sampling. It trains an SVDD model on each sample and incrementally develops data description for the training data. Stochastic subset solver provides a fast approximate description of the training data.

Table 3 shows the performance results for the single-SVDD model. The Botflows column shows a threshold value above which we consider the aggregated activity of a source device as an attack. Recall that a botflow is a NetFlow that is labeled as botnet-related. We now describe the columns of Table 3.

- True positive (TP): the number of device activities correctly classified as attack
- False positive (FP): the number of device activities incorrectly classified as attack
- True negative (TN): the number of device activities correctly classified as normal
- False negative (FN): the number of device activities incorrectly classified as normal
- True positive rate (TPR): the ratio $TP/(TP + FN)$
- False positive rate (FPR): the ratio $FP/(FP + TN)$

Note that the number of device activities that are considered as attack reduces with the increase in the botflows threshold value. For example, consider the first row from Table 3 with botflows threshold of 1. In this case, a device activity is considered an attack if the number of botflows in that device activity is greater than or equal to 1. The total number of such attack device activities is: $TP+FN = 164+39 = 203$. In the second row with botflows threshold of 2, the total number of attack device activities is: $TP+FN = 164+26 = 190$. The difference $203-190 = 13$ is the number of device activities with exactly 1 botflow, and are not considered as attack activities in the second row.

As Table 3 shows, the true positive rate increases with the botflows threshold. This is because the signal in the data increases with the number of botflows, and therefore the model's ability to correctly classify a device's activity as attack increases. The model correctly classifies all devices' activities in which the device sends nine or more botflows as attack.

Although the FPR in Table 3 is small (0.039), the number of false positives per true positive (FP/TP) is quite large (between 220–247). This is due to the issue discussed in Section 3. In the next section, we employ the proposed method from Section 4 on the data set.

5.4 The Multiple-SVDD Model Approach

Our evaluation employs KMeans clustering to cluster the CTU_NORM data set. In order to study the effect of cluster count on the results, we repeatedly apply the proposed method with cluster counts in the range 2–5. We train an SVDD model per cluster. Similar to the single-SVDD model, we use modified mean criterion method for bandwidth selection, and the stochastic subset solver. We score CTU_TEST using each of the cluster SVDD models. An observation is considered an outlier, that is, an attack, if the observation is an outlier per all of the cluster SVDD models. Otherwise, the observation is considered an inlier, that is, a non-attack.

As Table 4 shows, the FP/TP ratio decreases up to three clusters, and from four clusters onward it increases. Although the FP/TP ratio is minimum at three clusters, the true positive rate is lower compared to the other cases. As Table 3 shows, in case of single-SVDD model, at the botflows threshold of 9 the FP/TP ratio is 247, and the true positive rate is 1. From Table 4(a), at the botflows threshold of 9, the FP/TP ratio for the two-cluster case is 81, and the true positive rate is 0.99. As compared to the single-SVDD model, the FP/TP ratio is significantly smaller in the two-cluster case and the true positive rate is about the same. This shows that our approach leads to a significant reduction in false positives with a minimal impact on the true positive rate. Observe that the two-cluster case has 1 false negative whereas the single-SVDD model has 0 false negatives. The false negative can be addressed by increasing the botflows threshold from 9 to 10.

The optimal number of clusters using ABC criterion for the CTU_NORM dataset is 3. For cluster count of 3 and at the botflows threshold of 9, Table 4(b) shows that the true positive rate is 0.68, the number of false negatives is 47, and the FP/TP ratio is 17.99. Although the FP/TP ratio is smallest for the cluster count of 3, the false negatives are relatively higher. Therefore, instead of using the optimal cluster count, we suggest applying our approach with different cluster counts that are close to the optimal cluster

Botflows	TP	FP	TN	FN	TPR	FPR	FP/TP
1	152	11759	911868	51	0.749	0.013	77.36
2	152	11759	911881	38	0.800	0.013	77.36
3	152	11759	911887	32	0.826	0.013	77.36
4	152	11759	911889	30	0.835	0.013	77.36
5	152	11759	911905	14	0.916	0.013	77.36
6	145	11766	911906	13	0.918	0.013	81.14
7	145	11766	911906	13	0.918	0.013	81.14
8	145	11766	911916	3	0.980	0.013	81.14
9	145	11766	911918	1	0.993	0.013	81.14

(a) Two Clusters

Botflows	TP	FP	TN	FN	TPR	FPR	FP/TP
1	99	1781	921846	104	0.488	0.002	17.99
2	99	1781	921859	91	0.521	0.002	17.99
3	99	1781	921865	85	0.538	0.002	17.99
4	99	1781	921867	83	0.544	0.002	17.99
5	99	1781	921883	67	0.596	0.002	17.99
6	99	1781	921891	59	0.627	0.002	17.99
7	99	1781	921891	59	0.627	0.002	17.99
8	99	1781	921901	49	0.669	0.002	17.99
9	99	1781	921903	47	0.678	0.002	17.99

(b) Three Clusters

Botflows	TP	FP	TN	FN	TPR	FPR	FP/TP
1	140	3731	919896	63	0.690	0.004	26.65
2	140	3731	919909	50	0.737	0.004	26.65
3	140	3731	919915	44	0.761	0.004	26.65
4	140	3731	919917	42	0.769	0.004	26.65
5	140	3731	919933	26	0.843	0.004	26.65
6	133	3738	919934	25	0.842	0.004	28.11
7	133	3738	919934	25	0.842	0.004	28.11
8	133	3738	919944	15	0.899	0.004	28.11
9	133	3738	919946	13	0.911	0.004	28.11

(c) Four Clusters

Botflows	TP	FP	TN	FN	TPR	FPR	FP/TP
1	158	48154	875473	45	0.778	0.052	304.77
2	158	48154	875486	32	0.832	0.052	304.77
3	158	48154	875492	26	0.859	0.052	304.77
4	157	48155	875493	25	0.863	0.052	306.72
5	156	48156	875508	10	0.940	0.052	308.69
6	148	48164	875508	10	0.937	0.052	325.43
7	148	48164	875508	10	0.937	0.052	325.43
8	148	48164	875518	0	1.000	0.052	325.43
9	146	48166	875518	0	1.000	0.052	329.90

(d) Five Clusters

Table 4: Performance Results of SVDD Models

count, and selecting a cluster count that provides desired overall results in terms of: true positive rate, false negative count, and FP/TP ratio.

To further study the effect of cluster count, we extended the evaluation to cluster counts in range 6–10. As the cluster count is increased, the number of false positives increases but the true positive rate and false negative counts remain about the same as the five-cluster case.

6. RELATED WORK

Kenza et al. [13] show that a single SVDD model trained on data with disjoint clusters leads to a large number of false negatives (attack observations classified as normal). To address this problem, similar to our approach, they propose a hybrid SVDD and clustering approach in which they train a separate SVDD model for each cluster. Unlike Kenza et al., we show that a single-SVDD model, which is trained using an appropriate bandwidth can correctly model the data that contain multiple disjoint clusters, and it does not produce a large number of false negatives. In this paper we used modified mean criterion method for bandwidth selection, but other bandwidth selection methods based on the peak [12], mean [3] or the trace [4] criterion are equally capable of identifying the clusters. Section 3.2 demonstrates this result on the toy data set using the single-SVDD model. Further, the performance results for the single-SVDD model from Table 3 show that clustering is not needed for reducing the false negatives. A single-SVDD model is able to identify a majority of the attacks. We propose combining clustering with SVDD to reduce the number of false positives (the normal observations classified as attacks).

Görnitz et al. [8] propose *clusterSVDD*, a methodology that combines clustering and SVDD in a single formulation. The *clusterSVDD* method is based upon the primal formulation of SVDD as defined in Equation 1, but unlike our approach it lacks support for flexible data description that can be obtained using kernel functions. Our approach uses Gaussian kernel function for flexible data description that effectively models geometry of the training data.

Maglaras and Jiang [17] employ one-class support vector machine (OCSVM) to detect attacks on supervisory control and data acquisition (SCADA) systems. They employ normal network traffic from SCADA networks for training the OCSVM model and scoring traffic in real-time using the OCSVM model. In contrast to Maglaras and Jiang's work, we focus on detecting attacks in enterprise networks. We employ SVDD instead of OCSVM, which can be easily operationalized. In comparison to enterprise networks, the traffic in SCADA networks has less variety and variability leading to relative ease in anomaly detection. Therefore, techniques that are effective in detecting attacks in SCADA networks might not be effective in enterprise networks.

Umer et al. [23] evaluate various one-class classification techniques for extracting attacks from enterprise network activities. They categorize the techniques into three categories: density estimation, reconstruction methods, and boundary methods. The SVDD technique is included in the boundary methods. Similar to our paper, they employ CTU-13 data set for evaluation. They find that SVDD is more effective as compared to all of the other one-class classification techniques in detecting attacks. Our work goes beyond Umer et al.'s work by further improving the performance of the SVDD technique.

Lakhina et al. [15] employ principal components analysis (PCA) for detecting network-wide anomalies. PCA transforms data into a set of principal components that are or-

dered on the extent of the variance they explain in the data. Lakhina et al. separate the network traffic into normal and anomalous subspaces using PCA. The higher order principal components represent normal traffic, whereas the lower order components represent anomalous traffic. Unlike Lakhina et al.'s method, we employ the SVDD technique to construct a boundary around the normal traffic. Unlike PCA, SVDD can detect anomalies along any direction, hence it more effectively models real-world data sets.

7. CONCLUSION

This paper applies the SVDD technique for extracting cyber attacks from enterprise network activities. We show that a single-SVDD model for attack detection leads to a large number of false positives, especially when the training data set contains multiple disjoint clusters of different scale and density. To address this issue, we propose an approach that combines clustering with SVDD. We demonstrate the problem and the effectiveness of our proposed approach on a toy data set. Finally, we present an extensive evaluation of our proposed approach on the CTU-13 data set.

Note that although our proposed approach leads to lower false positives, it requires the additional step of clustering and associated tuning of the cluster count, which is not required by the single SVDD model. Our results confirm that the single-SVDD model has low false negatives (that is, it identifies a majority of the attacks). Therefore, in cases where processing capacity and time are limited and higher number of false positives is not an important issue, a single-SVDD model will be more appropriate.

Future Work

Our evaluation used a fixed hour-of-day time window for computing the features. If an attack straddles two consecutive hours, then the features computed using the fixed hour-of-day window may not be effective in detecting it. In future, we plan to investigate the impact of a sliding-time window and windows of different lengths on the effectiveness in attack detection.

In SVDD, all points on a distance value contour are classified as outliers without regard to the directionality. In some domains including cybersecurity, the directionality is relevant. For example, SVDD might classify a device that sends fewer bytes of data and communicates with fewer destination hosts (compared to the other devices) as an outlier. However, such activity of the device is not indicative of an attack. In future, we plan to study a variant of SVDD that considers directionality.

We also plan to develop an approach for incrementally updating a pre-trained SVDD model as new training data becomes available. This is relevant for enterprise settings in which the training data evolves continuously.

8. ACKNOWLEDGMENTS

The authors would like to thank Arin Chaudhuri for insightful discussions on SVDD, and Kathleen Walch for help in creating this manuscript.

9. REFERENCES

[1] F. Camci and R. B. Chinnam. General support vector representation machine for one-class classification of non-stationary classes. *Pattern Recognition*, 41(10):3021–3034, 2008.

[2] A. Chaudhuri, D. Kakde, M. Jahja, W. Xiao, S. Kong, H. Jiang, and S. Percdriy. Sampling method for fast training of support vector data description. In *Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–7. IEEE, 2018.

[3] A. Chaudhuri, D. Kakde, C. Sadek, L. Gonzalez, and S. Kong. The mean and median criteria for kernel bandwidth selection for support vector data description. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 842–849, Nov 2017.

[4] A. Chaudhuri, D. Kakde, C. Sadek, W. Hu, H. Jiang, S. Kong, Y. Liao, S. Peredriy, and H. Wang. The trace criterion for kernel bandwidth selection for support vector data description. *arXiv preprint arXiv:1811.06838*, 2018.

[5] D. A. R. P. A. (DARPA). *RFC-793: Transmission Control Protocol*, 1981.

[6] P. F. Evangelista, M. J. Embrechts, and B. K. Szymanski. Some properties of the Gaussian kernel for one class learning. In *Artificial Neural Networks–ICANN 2007*, pages 269–278. Springer, 2007.

[7] S. Garcia, M. Grill, H. Stiborek, and A. Zunino. An empirical comparison of botnet detection methods. *Computers and Security Journal*, 45:100–123, 2014.

[8] N. Görnitz, L. A. Lima, K.-R. Müller, M. Kloft, and S. Nakajima. Support vector data descriptions and k-means clustering: One class? *IEEE transactions on neural networks and learning systems*, 29(9):3994–4006, 2017.

[9] T. Hastie, R. Tibshirani, and J. Friedman. Unsupervised learning. In *The Elements of Statistical Learning*. Springer, 2009.

[10] C. S. Inc. *NetFlow Export Datagram Format*, 2007.

[11] S. I. Inc. *SAS Institute Inc SAS® Visual Data Mining and Machine Learning 8.3: Programming Guide: the SVDD procedure (Chapter)*, 2018.

[12] D. Kakde, A. Chaudhuri, S. Kong, M. Jahja, H. Jiang, and J. Silva. Peak Criterion for Choosing Gaussian Kernel Bandwidth in Support Vector Data Description. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM) (PHM2017)*, 2017.

[13] T. Kenaza, K. Bennaceur, and A. Labed. An efficient hybrid svdd/clustering approach for anomaly-based intrusion detection. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 435–443. ACM, 2018.

[14] S. Khazai, S. Homayouni, A. Safari, and B. Mojaradi. Anomaly detection in hyperspectral images based on an adaptive support vector method. *Geoscience and Remote Sensing Letters, IEEE*, 8(4):646–650, 2011.

[15] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *SIGCOMM Comput. Commun. Rev.*, 35(4):217–228, Aug. 2005.

[16] Y. Liao, D. Kakde, A. Chaudhuri, H. Jiang, C. Sadek, and S. Kong. A new bandwidth selection criterion for using svdd to analyze hyperspectral data. In

Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXIV, volume 10644, page 106441M. International Society for Optics and Photonics, 2018.

[17] L. Maglaras and J. Jiang. Intrusion detection in scada systems using machine learning techniques. In *Proceedings of Science and Information Conference (SAI)*, pages 626–631, Aug 2014.

[18] C. Sanchez-Hernandez, D. S. Boyd, and G. M. Foody. One-class classification for mapping a specific land-cover class: SVDD classification of fenland. *IEEE Transactions on Geoscience and Remote Sensing*, 45(4):1061–1073, 2007.

[19] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, pages 582–588, 2000.

[20] T. Sukchotrat, S. B. Kim, and F. Tsung. One-class classification-based control charts for multivariate process monitoring. *IIE Transactions*, 42(2):107–120, 2009.

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

[22] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[23] M. F. Umer, M. Sher, and Y. Bi. Applying one-class classification techniques to ip flow records for intrusion detection. *Baltic Journal of Modern Computing (BJMC)*, 5(1):70–86, 2017.

[24] A. Widodo and B.-S. Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6):2560–2574, 2007.

[25] Y. Xiao, H. Wang, L. Zhang, and W. Xu. Two methods of selecting Gaussian kernel parameters for one-class svm and their application to fault detection. *Knowledge-Based Systems*, 59:75–84, 2014.

[26] A. Ypma, D. M. Tax, and R. P. Duin. Robust machine fault detection with independent component analysis and support vector data description. In *Neural Networks for Signal Processing IX. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 67–76. IEEE, 1999.